



**UNIVERSIDAD CARLOS III DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**

**INGENIERÍA SUPERIOR EN INFORMÁTICA**

**PROYECTO FIN DE CARRERA**

**DISEÑO E IMPLEMENTACIÓN DE UNA  
CONSOLA DE COMUNICACIONES PARA EL  
METRO DE MADRID**

**Autor:**

Ignacio Ramírez-Cárdenas Montero de Espinosa

**Tutores:**

David Expósito Singh

Pablo Díaz Rebaque



Título:       Diseño e implementación de una consola de comunicaciones  
para Metro de Madrid

Autor:       Ignacio Ramírez-Cárdenas Montero de Espinosa

Director:     David Expósito Singh

Pablo Díaz Rebaque

#### EL TRIBUNAL

Presidente:   Javier Fernández Muñoz

Secretario:   Alejandro Calderón Mateos

Vocal:        Harith Al-Jumaily

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 5 de julio de 2010 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

#### VOCAL

SECRETARIO

PRESIDENTE



## **Agradecimientos**

Mi primer y más sentido agradecimiento es para mis padres, Diego Ramírez-Cárdenas Martínez y Ana María Montero de Espinosa Rojo. Sin vosotros, llegar hasta aquí habría sido imposible. Gracias por vuestro apoyo, ayuda y comprensión.

No me puedo olvidar de mis demás familiares, amigos y compañeros que también me ayudaron y me animaron en los momentos más difíciles y que han conseguido que pueda escribir ahora estas palabras.

Y por último, muchísimas gracias a Roberto Peña, por haber confiado en mí para una gran empresa en la cual he podido hacer grandes proyectos como el que nos ocupa. A Pablo Díaz Rebaque, por saber guiarme y enseñarme durante mi aprendizaje en el mundo laboral, y por supuesto, a David Expósito Singh por ayudarme y orientarme durante todo este proyecto, haciéndolo todo mucho más fácil.

De corazón, gracias a todos.





## Contenido

1. Introducción.....	14
1.1. Origen del proyecto.....	14
1.2. Descripción global en la que se encuadra el proyecto.....	16
1.3. Descripción del proyecto fin de carrera .....	18
2. Objetivos.....	20
2.1. Desarrollo de la aplicación .....	20
2.2. Implantación en un entorno real.....	20
2.3. Evaluación.....	23
3. Descripción de las tecnologías.....	24
3.1. Arquitectura hardware .....	24
3.2. Arquitectura software .....	28
• Python .....	28
• GLib .....	30
• D-Bus.....	31
4. Arquitectura propuesta .....	34
4.1. Descripción general .....	34
4.2. Diagrama de estados de la aplicación .....	35
4.3. Descripción de la interfaz gráfica .....	38
4.4. Otras funcionalidades.....	38
5. Requisitos.....	40
5.1. Funcionales.....	41
5.2. No funcionales.....	56
• Requisitos de interfaz de usuario .....	56
• Requisitos de interfaces externas.....	57
• Requisitos de prestaciones .....	57
• Requisitos de escalabilidad.....	58
6. Implantación .....	59
6.1. Problemas de implantación.....	59



6.2.	Soluciones aportadas.....	61
6.3.	Comentarios basados en la experiencia.....	65
7.	Evaluación del sistema.....	66
7.1.	Pruebas funcionales .....	66
7.2.	Casos de prueba .....	67
8.	Presupuesto .....	91
8.1.	Planificación y costes.....	91
8.2.	Análisis según COCOMO .....	92
	• Atributos de software .....	93
	• Atributos de hardware.....	94
	• Atributos de personal .....	94
	• Atributos de proyecto.....	95
8.3.	Análisis comparativo de costes .....	95
9.	Conclusiones .....	97
10.	Bibliografía .....	99
	Glosario .....	100





## Ilustraciones

Ilustración 1: Consola de Metro .....	21
Ilustración 2: Consola embarcada en la cabina .....	22
Ilustración 3: Detalle de la consola instalada .....	22
Ilustración 4: Rack de comunicaciones.....	23
Ilustración 5: Estructura general del sistema .....	24
Ilustración 6: Esquema de la consola de metro.....	26
Ilustración 7: Diagrama de clases de un ejemplo con D-Bus.....	33
Ilustración 8: Modelo Vista-Controlador implementado .....	34
Ilustración 9: Diagrama del sistema .....	35
Ilustración 10: Diagrama de flujo de la aplicación.....	36
Ilustración 11: Esquema de ejecución de los menús.....	37
Ilustración 12: Planificación inicial.....	91
Ilustración 13: Planificación real.....	91
Ilustración 14: Tren de la serie 2000 .....	98



## Tablas

Tabla 1: Tabla tipo para los requisitos .....	40
Tabla 2: Requisito RF_FUN-01 .....	41
Tabla 3: Requisito RF_FUN-02 .....	41
Tabla 4: Requisito RF_FUN-03 .....	42
Tabla 5: Requisito RF_FUN-04 .....	42
Tabla 6: Requisito RF_FUN-05 .....	42
Tabla 7: Requisito RF_FUN-06 .....	43
Tabla 8: Requisito RF_FUN-07 .....	43
Tabla 9: Requisito RF_FUN-08 .....	43
Tabla 10: Requisito RF_FUN-09 .....	44
Tabla 11: Requisito RF_FUN-10 .....	44
Tabla 12: Requisito RF_FUN-11 .....	44
Tabla 13: Requisito RF_FUN-12 .....	45
Tabla 14: Requisito RF_FUN-13 .....	45
Tabla 15: Requisito RF_FUN-14 .....	45
Tabla 16: Requisito RF_FUN-15 .....	46
Tabla 17: Requisito RF_FUN-16 .....	46
Tabla 18: Requisito RF_FUN-17 .....	46
Tabla 19: Requisito RF_FUN-18 .....	47
Tabla 20: Requisito RF_FUN-19 .....	47
Tabla 21: Requisito RF_FUN-20 .....	47
Tabla 22: Requisito RF_FUN-21 .....	48
Tabla 23: Requisito RF_FUN-22 .....	48
Tabla 24: Requisito RF_FUN-23 .....	48
Tabla 25: Requisito RF_FUN-24 .....	49
Tabla 26: Requisito RF_FUN-25 .....	49
Tabla 27: Requisito RF_FUN-26 .....	49
Tabla 28: Requisito RF_FUN-27 .....	50
Tabla 29: Requisito RF_FUN-28 .....	50
Tabla 30: Requisito RF_FUN-29 .....	50
Tabla 31: Requisito RF_FUN-30 .....	51
Tabla 32: Requisito RF_FUN-31 .....	51
Tabla 33: Requisito RF_FUN-32 .....	51
Tabla 34: Requisito RF_FUN-33 .....	52
Tabla 35: Requisito RF_FUN-34 .....	52
Tabla 36: Requisito RF_FUN-35 .....	52



Tabla 37: Requisito RF_FUN-36 .....	53
Tabla 38: Requisito RF_FUN-37 .....	53
Tabla 39: Requisito RF_FUN-38 .....	53
Tabla 40: Requisito RF_FUN-39 .....	54
Tabla 41: Requisito RF_FUN-40 .....	54
Tabla 42: Requisito RF_FUN-41 .....	54
Tabla 43: Requisito RF_FUN-42 .....	55
Tabla 44: Requisito RF_FUN-43 .....	55
Tabla 45: Requisito RF_FUN-44 .....	55
Tabla 46: Requisito RN_INT-01 .....	56
Tabla 47: Requisito RN_INT-02 .....	56
Tabla 48: Requisito RN_INT-03 .....	57
Tabla 49: Requisito RN_INT-04 .....	57
Tabla 50: Requisito RN_IEX-01 .....	57
Tabla 51: Requisito RN_PRE-01 .....	58
Tabla 52: Requisito RN_ESC-01.....	58
Tabla 53: Tabla tipo para los casos de prueba .....	67
Tabla 54: Caso de prueba CP_CONSOLA-01 .....	68
Tabla 55: Caso de prueba CP_CONSOLA-02 .....	68
Tabla 56: Caso de prueba CP_CONSOLA-03 .....	69
Tabla 57: Caso de prueba CP_CONSOLA-04 .....	69
Tabla 58: Caso de prueba CP_CONSOLA-05 .....	70
Tabla 59: Caso de prueba CP_CONSOLA-06 .....	70
Tabla 60: Caso de prueba CP_CONSOLA-07 .....	71
Tabla 61: Caso de prueba CP_CONSOLA-08 .....	71
Tabla 62: Caso de prueba CP_CONSOLA-09 .....	71
Tabla 63: Caso de prueba CP_CONSOLA-10 .....	72
Tabla 64: CP_CONSOLA-11 .....	72
Tabla 65: Caso de prueba CP_TECLADO-01 .....	72
Tabla 66: Caso de prueba CP_TECLADO-02 .....	73
Tabla 67: Caso de prueba CP_DISPLAY-01.....	73
Tabla 68: Caso de prueba CP_DISPLAY-02.....	73
Tabla 69: Caso de prueba CP_DISPLAY-03.....	74
Tabla 70: Caso de prueba CP_MICROA-01 .....	74
Tabla 71: Caso de prueba CP_MICROA-02 .....	75
Tabla 72: Caso de prueba CP_MICROA-03 .....	75
Tabla 73: Caso de prueba CP_RADIOTEL-01.....	76
Tabla 74: Caso de prueba CP_RADIOTEL-02.....	76



Tabla 75: Caso de prueba CP_RADIOTEL-03 .....	77
Tabla 76: Caso de prueba CP_RADIOTEL-04 .....	78
Tabla 77: Caso de prueba CP_RADIOTEL-05 .....	78
Tabla 78: Caso de prueba CP_RADIOTEL-06 .....	79
Tabla 79: Caso de prueba CP_RADIOTEL-07 .....	79
Tabla 80: Caso de prueba CP_RADIOTEL-08 .....	80
Tabla 81: Caso de prueba CP_RADIOTEL-09 .....	80
Tabla 82: Caso de prueba CP_RADIOTEL-10 .....	81
Tabla 83: Caso de prueba CP_RADIOTEL-11 .....	81
Tabla 84: Caso de prueba CP_MEGAFO-01 .....	82
Tabla 85: Caso de prueba CP_MEGAFO-02 .....	82
Tabla 86: Caso de prueba CP_MEGAFO-03 .....	82
Tabla 87: Caso de prueba CP_MEGAFO-04 .....	83
Tabla 88: Caso de prueba CP_MESTADO-01 .....	84
Tabla 89: Caso de prueba CP_MESTADO-02 .....	85
Tabla 90: Caso de prueba CP_MESTADO-03 .....	85
Tabla 91: Caso de prueba CP_MESTADO-04 .....	86
Tabla 92: Caso de prueba CP_MESTADO-05 .....	87
Tabla 93: Caso de prueba CP_MESTADO-06 .....	88
Tabla 94: Caso de prueba CP_MESTADO-07 .....	88
Tabla 95: Caso de prueba CP_MENUCON-01 .....	89
Tabla 96: Caso de prueba CP_MENUCON-02 .....	89
Tabla 97: Caso de prueba CP_MENUCON-03 .....	90
Tabla 98: Caso de prueba CP_MENUCON-04 .....	90
Tabla 99: Horas previstas asociadas al proyecto .....	92
Tabla 100: Horas reales asociadas al proyecto .....	92
Tabla 101: Factores de escala del proyecto .....	93
Tabla 102: Atributos de software .....	94
Tabla 103: Atributos de hardware .....	94
Tabla 104: Atributos de personal .....	95
Tabla 105: Atributos de proyecto .....	95





## **1. Introducción**

Este documento versa sobre el Proyecto de Fin de Carrera titulado “Diseño e implementación de una consola de comunicaciones para el Metro de Madrid” [1].

Se trata de un proyecto real, realizado para la empresa Metro de Madrid S. A., el cual actualmente ya se encuentra implantado. Este proyecto ha consistido en la creación de un nuevo sistema de comunicaciones para el parque móvil de dicha compañía.

### **1.1. Origen del proyecto**

Metro de Madrid es una empresa que tiene como principales funciones la explotación de las líneas de la red de Metro en funcionamiento, la planificación y mejora de la calidad del servicio de transporte y el mantenimiento y optimización de las instalaciones de la red del suburbano madrileño.

El objetivo fundamental de Metro de Madrid es el de ofrecer a los ciudadanos el mejor servicio de transporte público. Con el fin de conseguir la satisfacción de los ciudadanos, la Dirección de Metro de Madrid asume el compromiso de mejorar continuamente la calidad del servicio prestado.

Cierta parte de la flota de trenes de Metro posee ya cierta antigüedad. Estos trenes tienen ya algunos más de 20 años y carecen de sistemas informáticos y de sistemas de comunicaciones digitales. Por esto, y por necesidad de adaptarse a la nueva normativa, surgió la necesidad de actualizar dichos trenes.

Para ello se instalarán nuevos dispositivos de comunicaciones en los trenes así como un sistema informático que gestione las señales discretas que producen los sensores y demás elementos del tren, así como los nuevos sistemas de comunicación que se embarquen en los vehículos.

La necesidad inicial de este proyecto vino dada por la Dirección General de Telecomunicaciones, que instó a Metro de Madrid a la utilización de nuevos canales y frecuencias en su sistema de comunicación radiotelefónica, teniendo como plazo límite para el comienzo de esta utilización el 31 de diciembre de 2009.

En consecuencia, además de asegurar la correcta funcionalidad del equipo en sus diversas forma y condiciones, otra característica fundamental que ha de distinguir a este nuevo sistema es la fiabilidad, de tal manera que desde el origen del proyecto, con la realización de los diseños, cálculos, fabricación, implantación, pruebas, etc. se



tengan en cuenta todas las condiciones potenciales o reales, sobredimensionamientos, redundancias, calidades, etc. para que el presente equipo tenga una fiabilidad excelente y en cualquier caso inscrita en los valores exigidos por el cliente en su oferta.

Ha de recordarse que la radiotelefonía embarcada tiene la consideración en Metro de “equipo vital”, ya que históricamente la existencia de un solo agente por tren, se basaba, entre otras condiciones, en la disponibilidad y correcta funcionalidad del equipo de radiotelefonía que permitiese, en todo momento, la comunicación hablada en Tren (conductor) – Tierra (Puesto de Mando).

Se derivan de aquí graves repercusiones, en caso de disfunción de la radiotelefonía (avería del equipo, zonas de bajo nivel de señal, interferencias de cualquier tipo, imposibilidad de establecer comunicaciones habladas inteligibles, etc.), ya que en estas situaciones el personal de conducción podría solicitar el desalojo de viajeros del tren y la retirada del mismo de la línea, con todo lo que implica en cuanto a la merma en la calidad del servicio, costes de reparación, inmovilización del tren completo, influencia en la disponibilidad de la flota, etc..

Por otro lado, es sabido que la utilización cotidiana de la radiotelefonía embarcada es muy pequeña en cada tren (pese a la conexión permanente del equipo), limitándose al momento de la salida a línea desde el depósito, averías en línea, irregularidades en la señalización, etc. por lo que en principio se supone que no será dificultoso garantizar su utilización y correcto funcionamiento, en todo momento o circunstancia.

Así mismo, ha de tenerse en cuenta que la radiotelefonía y sus partes integrantes, son elementos manejados de una forma directa por todo el personal de conducción (microteléfono y su pulsador, teclados, etc.) razón por la cual su diseño será especialmente robusto incluso previendo un “mal trato” o “uso indebido”, utilización de indicaciones exteriores indelebles al uso cotidiano y a abrasión, respetando las condiciones ergonómicas exigidas por la legislación laboral y respetando la ubicación ya dispuesta para estos equipos en los trenes actuales.

Por esto, la empresa Metro de Madrid elaboró unos pliegos para crear un concurso público en el que se elaboraran ofertas para satisfacer estas necesidades del cliente.

Debido a este gran proyecto, la empresa Amper [2] tenía la oportunidad de continuar su larga asociación con este importante cliente, por lo que la empresa creó una oferta muy sugerente para Metro de Madrid, con importantes novedades y



mejoras, con la seguridad y el respaldo de ser una compañía importante y conocida por el cliente, por su solvencia, su alta eficiencia y la calidad de sus productos.

Amper es una compañía multinacional española líder en el diseño e implantación de sistemas de información y soluciones integrales de comunicaciones civiles y militares. Con una experiencia de más de cincuenta años en el sector de las Tecnologías de la Información y de las Comunicaciones ofrece productos y servicios de vanguardia que permiten cubrir las necesidades específicas de sus clientes. A lo largo de su historia, y gracias al profundo conocimiento del sector, Amper ha ido adaptándose para dar respuesta a las necesidades y retos específicos de un mercado cada vez más competitivo, complejo y exigente. Esta capacidad de adaptación ha llevado a la empresa a especializarse en segmentos donde gracias a su flexibilidad y capacidad innovadora, pueden ofrecer ventajas diferenciales a sus clientes.

En concreto, la unidad de negocio encargada de este proyecto es la de Homeland Security. Amper realiza proyectos de diseño e implantación de redes y sistemas propios para cubrir las necesidades de seguridad y comunicaciones críticas en grandes instalaciones, ciudades, comunidades autónomas y países. Es líder en sistemas de emergencia, control de fronteras y criptosistemas. Algunos de sus clientes más destacados son el Ministerio del Interior, Guardia Civil, ayuntamientos y grandes empresas como Metro de Madrid.

## **1.2. Descripción global en la que se encuadra el proyecto**

La necesidad del cliente es la renovación del equipamiento actual de los sistemas radio embarcados en los trenes. El nuevo equipamiento obviamente debe superar los niveles de fiabilidad, disponibilidad y mantenibilidad que estime el cliente como oportunos, además de ser compatible con el equipamiento actual. Todo esto implica:

- Desarrollo de un sistema radio compatible con los sistemas actualmente en explotación de forma que pueda ser utilizado como repuesto estándar. Dicho equipo presentará características duales de VHF y TETRA [3], y se ajustará a los requerimientos detallados por el cliente en el apartado de especificaciones técnicas que ofrece y será compatible con la infraestructura radio existente en sus instalaciones.
- Retirada del equipamiento antiguo (racks, consolas, antenas, cableado, etc.), siempre bajo la aprobación de Metro y con la máxima de seguir los procedimientos y las fechas que este indique, de forma que dicho cambio minimice su afección a la explotación.





- Instalación del nuevo equipamiento siempre bajo las premisas e indicaciones de Metro y con la aprobación del mismo.
- Verificar la instalación del sistema de radiotelefonía embarcado.

Por otro lado habrá de garantizar la calidad de la totalidad del equipamiento, tanto del que se renueve, como del que se mantenga, como de aquel que se opte por reciclar.

Este punto implica una revisión completa del equipamiento (estado de las consolas, racks, antenas, cableados, conectores, etc.) y la emisión de los certificados oficiales en los que se especifique el estado de los mismos y se garantice su correcto funcionamiento en base a los parámetros de fiabilidad, disponibilidad y mantenibilidad marcados.

Por último habrá de adecuar el parque de equipos radio, sistemas embarcados en trenes y terminales móviles (walkie-talkies), tanto del equipamiento que se renueva, como del que se mantiene, a la normativa vigente, para ajustarse a las nuevas frecuencias asignadas a Metro de Madrid, llevando a cabo además su ampliación en dos canales. Esto supone como mínimo:

- Reprogramación de todos los terminales móviles (walkie-talkies VHF).
- Reprogramación de todas las radios embarcadas en trenes.
- Reprogramación de las consolas y racks para la introducción en los menús de selección de los nuevos canales.

Para este gran y diversificado proyecto, en la parte que nos ocupa, se dispuso de un tiempo estimado de unos **diez meses** para el diseño del hardware y el desarrollo de la aplicación. Después se tendría un mes para las pruebas finales y otro para la producción en serie de todo el sistema hardware y la implantación definitiva en los trenes.

Para este proyecto, por tanto, primero hubo de asignarse un equipo de hardware que diseñara la placa que iba a llevar la consola del tren, así como la destinada al rack de comunicaciones que gestionaría todas las señales e información que generaran los terminales. Además de esto, hubo que diseñar la estructura en la que irían integrados todos estos componentes junto con la CPU del sistema.

Esta parte fue llevada a cabo por un equipo de entre dos y tres personas dedicadas a tiempo completo durante un periodo de unos tres meses y medio.



Por otro lado, y en paralelo, otra persona se encargó de crear los drivers necesarios para los componentes de este sistema.

Transcurridos tres meses del inicio del proyecto, empezó el trabajo del equipo de desarrollo software. Estaba compuesto por dos personas a tiempo completo. Una de ellas era la encargada del desarrollo de todo el controlador y la parte inteligente del rack de comunicaciones. Además contaba con otra persona de apoyo que realizó modificaciones en los sistemas de radio para adaptarlos a las especificaciones necesarias.

El proyecto que nos ocupa, estaba basado en la otra parte del desarrollo software, que era la encargada de llevar toda la gestión de la consola del tren, que sería la parte con la que interactuaría finalmente el usuario final del sistema. Las tareas a realizar serán las siguientes:

- Control del display gráfico de la consola.
- Control del teclado de la consola.
- Control de los leds de la consola.
- Creación de los menús de usuario.
- Desarrollo de los requisitos específicos de la consola.

### **1.3. Descripción del proyecto fin de carrera**

El Proyecto de Fin de Carrera “Diseño e implementación de una consola de comunicaciones para el Metro de Madrid” se encuadra en el entorno anteriormente descrito.

Concretamente, este proyecto está dentro del desarrollo del nuevo sistema de comunicaciones, para el cual se desarrollará una consola que sustituirá las existentes previamente. Esta consola constará de un microteléfono y un sistema de manos libres para poder llevar a cabo las conversaciones, así como de un display para mostrar información y de un teclado para que el usuario pueda introducir datos y realizar comandos.

Esta consola irá conectada a un rack de comunicaciones que será con el que haya que interactuar para mandar comandos e información al puesto de mando u otros trenes, así como para recibir eventos del estado de los trenes o información sobre las comunicaciones.



Para gestionar esta consola, por lo tanto, habrá que generar un programa que interactúe con los drivers de los elementos hardware que componen dicha consola que están creados al efecto.

Además de esta interacción con el hardware de la consola, también habrá que desarrollar la parte de la interfaz de usuario con la que el maquinista interactuará. Esta aplicación consiste en una serie de menús que de una manera gráfica visualizarán los datos provenientes del rack de comunicaciones. Además de esto, también tendrán que desarrollarse funcionalidades que son ofrecidas directamente a través del teclado, como son la gestión de las dos radios existentes, la analógica VHF y la digital TETRA, así como comandos que se envían al puesto de mando.

Además de esto, también se crearán puertas traseras en el sistema para labores de mantenimiento por parte de gente cualificada en la actualización y reparación de estos sistemas.

Por otra parte, y como se ha descrito anteriormente, este sistema ha de ser altamente fiable, por lo que otra sección importante del proyecto consistirá en las pruebas de rendimiento y de fiabilidad que se llevarán a cabo para la correcta validación del sistema por parte del cliente como se verá más adelante en los casos de prueba.

Para ello, se desarrollará un plan de pruebas minucioso que permitirá comprobar la correcta funcionalidad del sistema, así como la integración con los demás sistemas existentes en los trenes.



## **2. Objetivos**

Durante este proyecto se han alcanzado diversos objetivos que se explican a continuación.

### **2.1. Desarrollo de la aplicación**

El primero y más destacable consiste en la aplicación que gestiona la consola.

Consiste en una aplicación escrita en Python que llevará el control de los dispositivos de entrada y salida que ofrece la consola de tren. Los dispositivos de entrada son el teclado, el sensor de luz y las señales que crea el microteléfono, aunque la aplicación interactuará indirectamente con estas últimas. Los dispositivos de salida, por el contrario, son el display, y los leds y el altavoz.

Esta aplicación se comunicará con el resto del sistema mediante un bus de datos lógico. Para usar este bus de datos habrá que implementar una interfaz de uso para poder obtener la información deseada para mostrársela al usuario, o por el contrario para mandar datos u órdenes que se introduzcan, para que el demonio que gestiona este bus se encargue de hacer con ella lo que estime conveniente.

A parte de estas funcionalidades, esta aplicación también se encargará de llevar el estado del tren, puesto que si pasa a estar en modo inactivo deberán cortarse las comunicaciones y desactivarse los sistemas asociados, como la pantalla, la megafonía y demás, quedándose en un modo latente esperando la nueva activación de los mismos mediante una única señal que será la que pueda recibir el sistema estando en este estado.

### **2.2. Implantación en un entorno real**

Este sistema se ha de instalar, inicialmente, en los trenes de la serie 2000 [4]. Estos trenes, los más antiguos que posee el parque móvil de Metro de Madrid, carecen de sistemas digitales, por lo que habrá de reemplazar los antiguos sistemas analógicos por los nuevos desarrollados en este proyecto.

La consola de comunicaciones será la parte visible del sistema, y será con la que interactúe directamente el maquinista. Esta consola dispone de una display gráfico que mostrará información al usuario, unos leds que indicaran de manera rápida el estado del sistema de comunicaciones, un teclado con el que realizar operaciones o introducir datos y por último un microteléfono y un sistema de manos libres con el que el maquinista podrá comunicarse con otros terminales.

La consola desarrollada se muestra a continuación.



**Ilustración 1: Consola de Metro**

Esta consola ira embarcada en el panel de controles que manejará el maquinista. Se encuentra en una posición accesible que facilitará su visión y uso en condiciones de baja visibilidad como es normalmente el entorno en el que se utilizará el sistema.





Ilustración 2: Consola embarcada en la cabina

Como se puede observar en la siguiente ilustración el sistema va completamente encajonado en la cabina, por lo que hubo de diseñarse a medida, con las especificaciones del cliente, para que se pudiera adaptar a los sistemas existentes previamente en los trenes.



Ilustración 3: Detalle de la consola instalada

Por último, el rack de comunicaciones, ira instalado en un armario de comunicaciones que posee el tren. Al igual que la consola, tuvo que ser hecho a medida para poder encajar en estos armarios.



Ilustración 4: Rack de comunicaciones

### 2.3. Evaluación

Además de desarrollar e instalar este sistema se creará un plan para verificar la funcionalidad y la usabilidad del sistema para el cliente.

Todo sistema debe cumplir con los requisitos que han sido adquiridos por el cliente. Estos requisitos tienen que describir exactamente lo que el cliente quiere que el sistema haga y tienen que ser probados en las diferentes fases del proyecto para asegurar que el progreso y la calidad del producto cumplan con las expectativas que tiene el cliente.

### 3. Descripción de las tecnologías

En este apartado se van a describir las tecnologías usadas para este proyecto, tanto a nivel hardware, como a nivel software.

#### 3.1. Arquitectura hardware

La consola de tren es un sistema informático que brinda la posibilidad a los conductores de comunicarse con varios elementos del mismo tren y de la infraestructura de Metro. La consola integra las comunicaciones de voz y datos del tren.

Se compone básicamente de dos elementos, una consola de visualización y un rack oculto de comunicaciones.

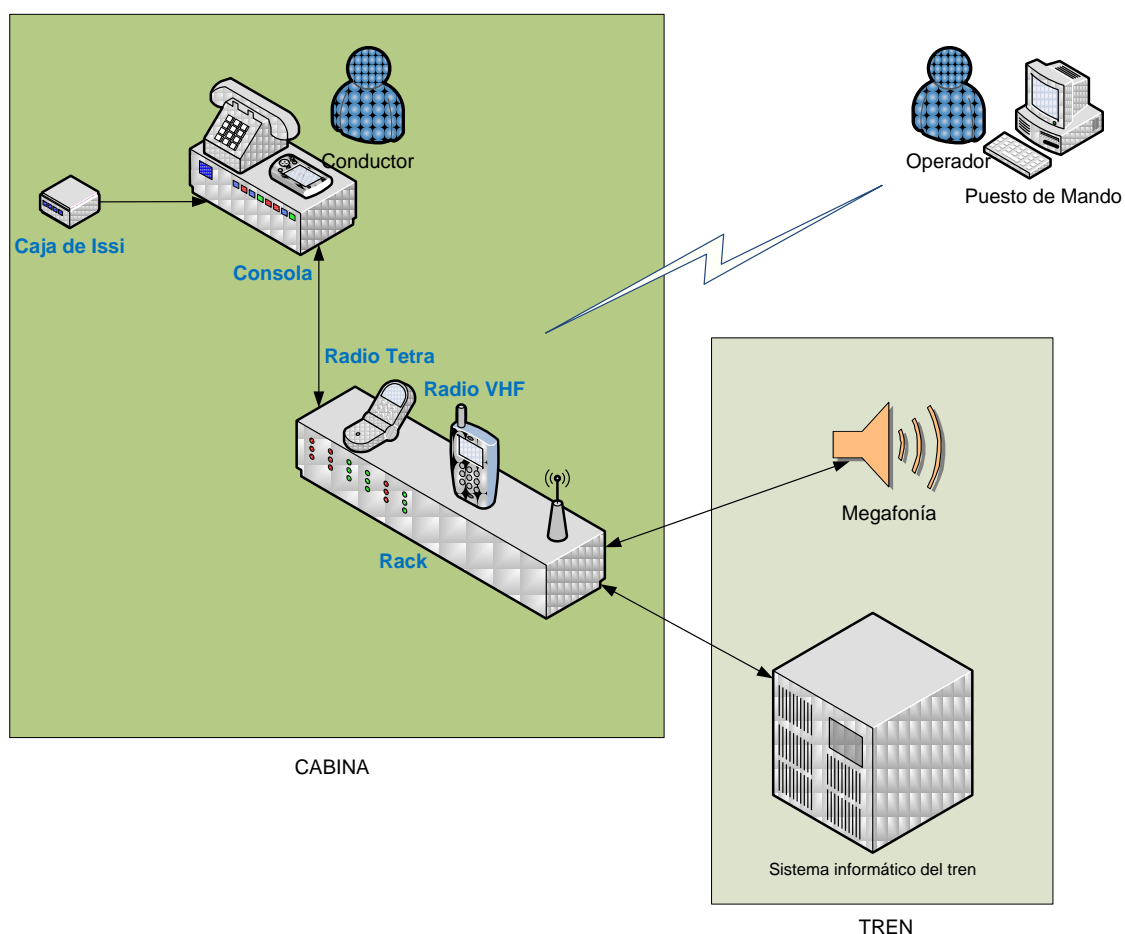


Ilustración 5: Estructura general del sistema

El rack es la parte inteligente del sistema, dónde se ubica el software y el hardware que harán posibles las comunicaciones del conductor con el resto del sistema.





En el rack se encuentran los sistemas de comunicación del tren con el puesto de mando y los demás trenes. Contiene dos radios, una VHF (analógica) y otra UHF (Tetra digital). La radio VHF se controla mediante E/S digitales de la propia CPU y la radio Tetra se controla utilizando un puerto serie RS-232. Además de esto se encuentra una caja de issi que permite identificar unívocamente la radio TETRA.

El rack del tren se comunica con varios sistemas externos mediante distintos elementos, que dependiendo de la serie o modelo del tren, funcionarán de una forma o de otra. Estos sistemas externos son:

- **SICAS o sistema informático del tren:** Se trata del “ordenador de a bordo” del tren, permite interactuar a la consola con el resto de los sistemas así como conocer el estado de los mismos. Este elemento dará más o menos información en función de la versión instalada en la serie de trenes en que se encuentre. Puede ocurrir que no haya sistema informático en cuyo caso se utilizarán para la información ciertas señales discretas que proporciona el tren.
- **Sistema de megafonía:** El sistema de megafonía permite emitir mensajes de voz desde la consola hasta el conjunto de los vagones del tren. También se utiliza para permitir la comunicación con los vagones en caso de emergencia. El sistema de megafonía es un sistema hardware que proporciona una especificación eléctrica y de señalización.
- **Puesto de mando:** A nivel de datos la consola del tren se comunica con el puesto de mando mediante mensajes SDS usando la radio Tetra.

Para controlar el rack, se dispone de un SBC (Single Board Computer) que además de contener la CPU en sí misma contiene muchos periféricos propios de un PC así como la posibilidad de expansiones por el bus PC/104 (similar al bus local de un PC).

Esta tarjeta es la que contendrá el software del sistema y hará las veces de central de control del resto de tarjetas y periféricos.

Cabe destacar las siguientes características:

- **Procesador:** La placa contiene un procesador de tipo PC (AMD Geode) de 500MHz con refrigeración pasiva. Esta potencia se ha estimado como suficiente para el conjunto de las acciones que se realizarán.
- **Memoria:** La tarjeta tiene 512MB de memoria en modulo y varios puertos de expansión para memoria secundaria.
- **Disco duro:** Se utilizara la expansión para tarjetas CompactFlash para almacenar todo el software, los datos y los logs de las aplicaciones.

- **Sistema operativo:** La complejidad del software para este sistema requiere que se instale un sistema operativo completo. La opción elegida es una distribución de Linux basada en Ubuntu, pero compilada específicamente para este proyecto, para que sea capaz de gestionar todos los periféricos de la tarjeta correctamente.
- **Watchdog:** La placa viene con un elemento hardware para vigilar el status de la misma, además, Linux implementa mecanismos para utilizar este hardware para vigilar servicios y demás tareas con el fin de garantizar la disponibilidad del sistema.

La consola de visualización es el elemento de interfaz humana que utiliza el conductor para comunicarse con los distintos elementos externos.

De cara al rack la consola no es más que un periférico de entrada (teclado, señales del microteléfono, etc.) y de salida (display, leds, altavoz).



Ilustración 6: Esquema de la consola de metro

Es esta consola en la que se centra el proyecto.



Por un lado dispone de un display gráfico monocromo con retro iluminación. Este display es el encargado de mostrar la información al maquinista, por lo tanto se tuvo que estudiar detenidamente la manera de mostrar la información de una manera directa y rápidamente legible. Además de esto hay que controlar la retro iluminación, puesto que el sistema tiene que ser duradero y la luz no es conveniente tenerla encendida siempre, pero al ser un entorno oscuro es imprescindible para poder visualizar los datos. Otro parámetro a controlar del display es su contraste puesto que dependiendo de la luminosidad que haya o incluso de la antigüedad del display, puede ser necesario modificarlo.

Además otro dispositivo de salida de información son los leds. Estos leds, muestran de una manera rápida y eficaz el estado de la radio y el microteléfono sin que el conductor tenga que atender al display. Para distinguir los cinco leds disponibles, a parte del letrero indicador, cada uno dispone de un color diferente. Además de estos, se encuentra también el led de encendido y apagado del teclado. Este, sin embargo dispone de tres estados, apagado, en reposo o iniciando y encendido.

Por otro lado, dispone de dos dispositivos de entrada que debían controlarse. El primero de ellos es el teclado. Se trata de un teclado especial de larga duración, en el que las teclas no funcionan presionando, si no por contacto. Es un sistema muy sensible que hubo que controlar muy cuidadosamente pues es fundamental evitar pulsaciones involuntarias del mismo, puesto que podría enviar señales incorrectas. Además de esto, tuvo que implementarse la posibilidad de ofrecer varias pulsaciones simultáneas de cara a ofrecer una puerta trasera para labores de mantenimiento.

Además de ello existe un sensor de luz para controlar la luminosidad del teclado. Al igual que ocurre con el display, este sistema se encuentra en un entorno oscuro y hay que facilitar al maquinista la lectura del teclado. No obstante, y por los mismos motivos de durabilidad hubo que controlar la luminosidad disponible en todo momento para apagar la retro iluminación en caso de pasar de cierto umbral de luminosidad.

En cuanto al diseño hardware la consola es un elemento sencillo de lógica programable que permite interactuar con los distintos periféricos. El elemento controlador de la consola es la propia CPU que se comunica mediante el módem de línea con el hardware de la consola.



Para conseguir esta abstracción se han desarrollado una serie de drivers que permiten interactuar con los elementos hardware del rack así como los de la consola, abstrayéndonos en todo momento del protocolo de comunicaciones con la misma.

### 3.2. Arquitectura software

Para desarrollar este proyecto, se han usado varias tecnologías software como se indican a continuación:

- **Python**

Dentro de este proyecto existen multitud de factores a tener en cuenta a la hora de tomar decisiones en la arquitectura de la misma. Uno de los factores restrictivos es el display en el que se muestra la información, ya que no es un monitor al uso, y lo que muestra únicamente son imágenes, no texto. Por lo tanto se estudiaron diversas maneras para poder interactuar con él.

Una de ellas era desarrollar la interfaz de la aplicación en C o C++ al igual que el resto del sistema, y crear un Frame Buffer para incrustar en él las imágenes que se han de mostrar. Sin embargo, finalmente consiste en tratar con imágenes y este procesado sería bastante complejo debido a las librerías existentes en estos lenguajes. Por lo tanto, se planteó la opción de usar Python [5], un lenguaje con multitud de bibliotecas útiles para esta tarea, y además es compatible con las demás tecnologías que se usarán para desarrollar el resto del sistema como se verá más tarde.

Python es un lenguaje de programación de alto nivel de propósito general cuya filosofía de diseño hace hincapié en la legibilidad del código. Python tiene como objetivo aunar la sintaxis muy clara con su biblioteca estándar amplia y completa. Prueba de ello es el uso de la sangría para el bloque de delimitadores que es inusual entre los lenguajes de programación populares.

Python soporta múltiples paradigmas de programación, principalmente, aunque no limitado, orientado a objetos, imperativo y funcional, y dispone de un completo tipo dinámico y automático sistema de gestión de memoria. Al igual que otros lenguajes dinámicos, Python se utiliza frecuentemente como un lenguaje de script, pero también se utiliza en otra amplia gama de contextos

Python es un lenguaje que está disponible para los principales sistemas operativos: Windows, Linux / Unix, OS / 2, Mac, Amiga, entre otros. Y su código fuente se ejecutará sin problema en cualquiera de ellos.



Las características de este lenguaje son las siguientes:

- **Propósito general:** Se pueden crear todo tipo de programas. En general, puede ser usado para cualquier desarrollo. Para ciertas tareas más comunes, existen librerías para facilitar la programación que permiten la reutilización de código.
- **Multiplataforma:** Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.
- **Interpretado:** Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos *bytecodes* que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.
- **Interactivo:** Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.
- **Orientado a Objetos:** La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- **Funciones y librerías:** Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de cadenas, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.
- **Sintaxis clara:** Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave `begin` y `end`. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.



- **GLib**

El sistema que se ha creado es asíncrono y se necesita un bucle de eventos para generar los eventos que son necesarios para que los distintos procesos se ejecuten convenientemente. Había previamente desarrollados otros bucles de eventos dentro de la empresa, pero sin embargo fueron desechados por su complejidad y por la falta de compatibilidad con Python, lenguaje necesario, como se ha explicado anteriormente, para desarrollar la interfaz de la consola.

Glib [6] es una biblioteca de propósito general que se usa para implementar muchas funciones no gráficas. Es una biblioteca de bajo nivel, siendo la estructura básica de GTK+ y GNOME. Proporciona manejo de estructura de datos para C, portabilidad, interfaces para funcionalidades de tiempo de ejecución como ciclos, hilos, carga dinámica o un sistema de objetos.

Uno de los mayores beneficios de usar GLib es que provee una interfaz de plataforma independiente que permite que el código pueda ser usado en diferentes sistemas operativos. Otro aspecto de GLib es la amplia gama de tipo de datos que deja disponible al desarrollador.

GLib provee tipos de datos en C que usualmente se incluyen de forma estándar a otros lenguajes, como por ejemplo listas enlazadas. Otros tipos de datos básicos incluidos son colas doblemente enlazadas, árboles binarios auto balanceados, etc.

Las cadenas de caracteres en GLib son similares a las de C++, porque son buffers de texto que crecen automáticamente cuando se agregan datos.

Los segmentos de memoria son una forma eficiente de crear secciones de memoria que tienen todos los mismos tamaños. Pueden ser usados para crear vectores aún de elementos de tamaño conocido.

GLib además de proporcionar varios tipos de datos, también dispone de numerosos tipos de funciones. Se encuentran funciones de manipulación de archivos, soporte de internacionalización, cadenas de caracteres, advertencias, banderas de depuración, carga dinámica de módulos, sólo por nombrar algunas.

Otra utilidad y la principal por la que se escogió GLib es su bucle de eventos. El bucle de eventos principal gestiona todas las fuentes disponibles de eventos para GLib y aplicaciones GTK+. Estos eventos pueden venir de cualquier número de diferentes tipos de fuentes tales como los descriptores de archivos (archivos planos, las tuberías o



sockets) y tiempos de espera. Los nuevos tipos de orígenes de eventos también se pueden agregar.

Para permitir que varios conjuntos independientes de las fuentes que se manejan en diferentes hilos, cada fuente se asocia con un `GMainContext`. Un `GMainContext` sólo se puede ejecutar en un hilo, pero las fuentes pueden añadirse y excluirse del mismo por otros subprocesos.

A cada origen de los eventos se les asigna una prioridad. La prioridad predeterminada es 0. Los valores menores que 0 denotan una mayor prioridad. Los valores mayores que 0, por el contrario, denotan menor prioridad. Los eventos provenientes de fuentes de alta prioridad siempre se procesan antes de los acontecimientos que surgen de fuentes de menor prioridad.

Funciones de reposo también se pueden añadir, y se les asignará igualmente una prioridad. Estos se llevarán a cabo cada vez que no haya eventos con mayor prioridad que estén listos para ser procesados.

El tipo de datos `GMainLoop` representa el bucle de eventos principal. Un `GMainLoop` se crea con `g_main_loop_new()`. Después de añadir los orígenes de los evento iniciales, se llama a `g_main_loop_run()`. Esto comprueba continuamente si existen nuevos eventos de cada una de las fuentes y los lanza. Finalmente, el procesamiento de un evento que ha producido alguna de las fuentes conduce a una llamada a `g_main_loop_quit()` para salir del bucle principal.

Es posible crear nuevas instancias de `GMainLoop` recursivamente. Esto es a menudo utilizado en aplicaciones GTK+ al mostrar cuadros de diálogo modal. Hay que tener en cuenta que los orígenes de los eventos están asociados con un `GMainContext` particular, y se comprobarán y lanzarán en todos los bucles principales asociados con ese `GMainContext`.

GTK+ contiene las envolturas de algunas de estas funciones, por ejemplo, `gtk_main()`, `gtk_main_quit()` y `gtk_events_pending()`, que serán las usadas en esta aplicación.

- **D-Bus**

Por último, falta comunicar los distintos procesos entre sí. Se plantearon diversas maneras de conseguir esto, como implementar un nuevo protocolo basado en sockets, o reutilizar protocolos ya creados por la empresa previamente. Sin embargo,



estos últimos se basaban más en la comunicación entre máquinas que en la comunicación entre procesos.

Además de estas opciones descritas, también existía la opción de usar D-Bus [7].

D-Bus es un sistema para comunicar procesos. Su arquitectura dispone de varias capas:

- La librería `libdbus` que permite a dos aplicaciones conectarse la una a la otra para intercambiar mensajes.
- Un demonio de bus de mensajes, construido sobre `libdbus`, al que se pueden conectar múltiples aplicaciones. El demonio puede enrutar los mensajes de una aplicación hacia otra.
- Librerías que envuelven a ciertos frameworks, como `libdbus-glib` y `libdbus-qt`. También hay enlaces a otros lenguajes como Python. Estas son las librerías que la mayoría de la gente debe usar pues simplifican la programación con D-Bus.

El diseño de D-bus se basa en dos casos específicos:

- Comunicación entre dos aplicaciones de escritorio de la misma sesión, de manera que se traten como a un conjunto
- Comunicación entre la sesión de escritorio y el sistema operativo, incluyendo normalmente al núcleo y a diversos procesos y demonios del sistema.

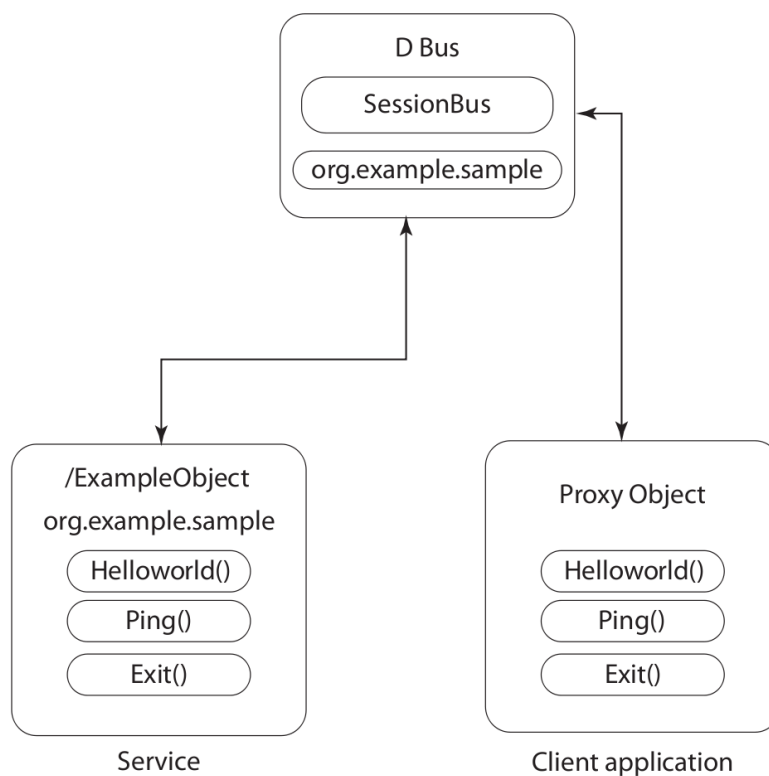
Toda aplicación que utilice D-Bus contiene objetos, normalmente definidos (aunque no es necesario) como `GObject`, `QObject`, objetos C++ u objetos Python. Cuando se recibe un mensaje a través de una conexión D-Bus, éste se envía a un objeto específico, no globalmente a la aplicación. De esta forma, D-Bus se apoya en componentes software, y de cara al usuario parece como si un objeto fuera serializado a través de la conexión creada, sin importar si existe o no un objeto al otro lado de la comunicación.

Para permitir la especificación del destino a los mensajes, debe existir un medio para indicar esa referencia. En muchos lenguajes de programación, esto recibe el nombre de puntero o referencia. Sin embargo, estas referencias se implementan como direcciones de memoria relativas al espacio de direcciones de la aplicación, lo cual no puede intercambiarse de una aplicación a otra.



Para resolver esto, D-Bus introduce un nombre para cada objeto. Dicho nombre se muestra como una ruta del sistema de ficheros. Por ejemplo, un objeto podría llamarse */org/freedesktop/DBus/Object*.

Un ejemplo básico para entender esta arquitectura con D-Bus sería el siguiente:



**Ilustración 7: Diagrama de clases de un ejemplo con D-Bus**

Aquí tendremos en el servidor un objeto sencillo llamado `ExampleObject` que derivará de `dbus.service.object`, que consistirá de los métodos, `HelloWorld()`, `Ping()` y `Exit()`. Estos métodos tendrán que ser decorados de una manera estándar para que sean identificados correctamente por todos los clientes conectados al D-Bus. Esto se hace mediante `@dbus.service.method("org.example.sample", in_signature = '', out_signature = '')` siendo el primer parámetro el nombre del objeto, el segundo la entrada de datos que recibe el método y el tercero la salida que tiene dicho método.

Por otro lado en el lado del cliente lo que se ha de hacer es crear un bus de sesión, obtener el objeto que queremos, en este caso `"org.example.sample"` y crear una interfaz de acceso al mismo. Una vez hecho esto solo tendremos que acceder a los métodos que ofrece dicha interfaz como ocurriría en un caso normal, abstrayéndonos por tanto donde se encuentra el objeto que nos ofrece ese servicio.

## 4. Arquitectura propuesta

En este apartado veremos la arquitectura finalmente implementada para realizar el proyecto solicitado por Metro de Madrid.

### 4.1. Descripción general

El software que se implementará en el rack seguirá el esquema de Modelo-Vista-Controlador:

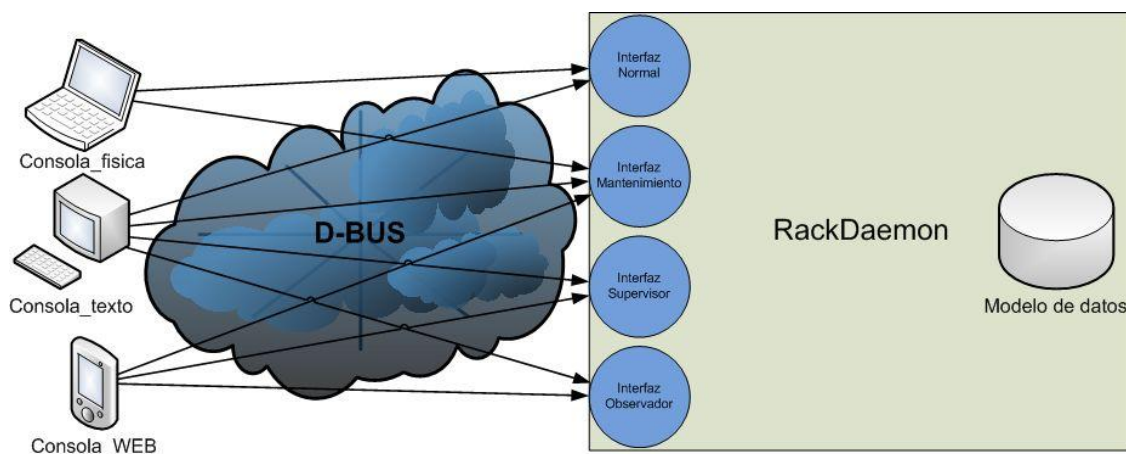


Ilustración 8: Modelo Vista-Controlador implementado

- La aplicación principal (el controlador) es un servicio que gestionará las comunicaciones del sistema siendo el único elemento que acceda al modelo de datos y a los distintos dispositivos hardware de comunicaciones.
- Las distintas vistas dependerán del dispositivo que se utilice para su visualización. La más importante es la que muestra los datos en la consola física o consola de visualización (interfaz con el conductor).
- Las interfaces del sistema son implementadas por el RackDaemon y representan los distintos niveles de acceso (según permisos) hacia el modelo de datos y los dispositivos.

En este desarrollo se ha generado el programa “ConsolaMain” que interactúa con el conductor del tren mediante la consola de visualización para transmitir las acciones al “RackDaemon” que las hará efectivas.

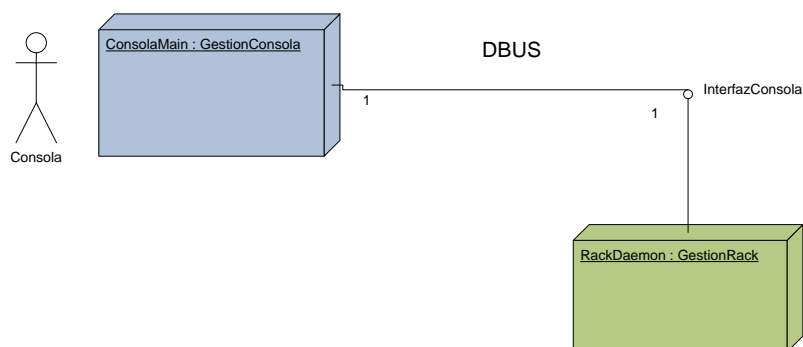


Ilustración 9: Diagrama del sistema

El objetivo del ConsolaMain es controlar el display y los leds de la consola así como responder a las acciones del teclado con comandos hacia el RackDaemon. Algunas señales del tren también pueden provocar cambios en la consola, por tanto, estas señales (que son publicadas por el RackDaemon) deben ser tratadas en el ConsolaMain para tomar acciones a tal efecto.

Para conseguir esto se ha creado una interfaz de acceso entre el demonio existente en el rack para poder obtener cierta información sobre el sistema para ser mostrada propiamente.

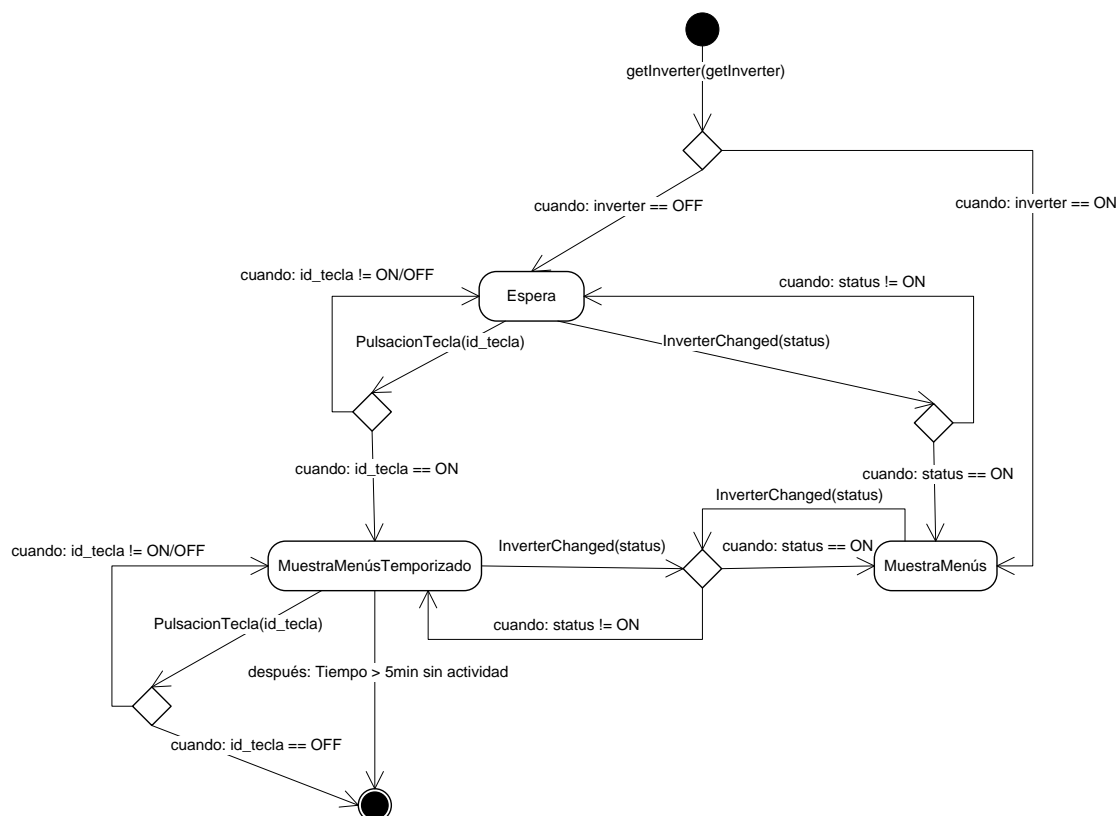
Además de esto, por ser un sistema asíncrono, se estipularon una serie de señales que ofrecería el rack que debería de capturar el programa para poder actualizar la información pertinente en la consola.

## 4.2. Diagrama de estados de la aplicación

El programa ConsolaMain consistirá en un bucle de ejecución basado en la librería GLIB que proporciona una capa de abstracción para D-Bus.

De cara a la consola existe un evento principal, que es el de la activación de la cabina del tren. Es imprescindible que la consola no se pueda apagar cuando este encendido el tren, y por otro lado, ha de apagarse transcurrido un tiempo si se ha encendido estando parado el tren.

A continuación se muestra el diagrama de estado de alto nivel que define el flujo de ejecución del programa:



**Ilustración 10: Diagrama de flujo de la aplicación**

Mientras nos encontremos en los estados MuestraMenús y MuestraMenúsTemporizados se debe seguir el siguiente esquema de ejecución dentro de los menús:





### 4.3. Descripción de la interfaz gráfica

La pantalla en la que se visualiza la información no es un monitor corriente, por lo que se ha tenido que desarrollar un sistema para plasmar imágenes en ella. En este display hubo de representarse la información de la manera más compacta, coherente y accesible que fuera posible, pues por problemas de legibilidad se limitó a cinco líneas el tamaño de la pantalla.

La pantalla principal, constará de una línea superior informativa, que constará de la hora actual, así como de diversos iconos según el estado de la consola, además de mostrar la cobertura.

El resto de menús dispondrán también de una línea superior también con el título del menú para indicar al usuario donde se encuentra.

Por último, las pantallas que así lo requieran, y tengan la quinta línea libre, dispondrán de una ayuda contextual, para facilitar la operación al maquinista.

### 4.4. Otras funcionalidades

A lo largo del proyecto se han realizado diversas tareas en el desarrollo de esta aplicación, las cuales se describen a continuación:

- **Conexión D-Bus:** La información mostrada, y la recibida se transmite a través de D-Bus, un bus lógico de datos basado en el paso de mensajes. Tras conectarse a él, hubo que desarrollar un protocolo con unas interfaces claramente definidas para entenderse con el demonio con el que se comunica el programa.
- **Bucle eventos GLib:** Para poder interactuar con el teclado se usó la librería de GLib que posee un bucle de eventos que gestiona para recibir las pulsaciones realizadas sobre el teclado.
- **Gestión de eventos:** Debido a la escasa memoria disponible y optimizar de la mejor manera el código se creó una clase de gestión de eventos para que la clase encargada de comunicarse con el bus de datos.
- **Iconos:** Una de las mejoras importantes de este producto respecto al anterior existente era la mejora del display, puesto que el nuevo pasaba a ser gráfico. Por ello, hubo que crear iconos para que se pudiera visualizar la información de una manera más rápida y cómoda.
- **Leds:** La consola dispone de una serie de leds para indicar el estado de la misma. Esta aplicación por tanto gestiona los mismos para mostrar correctamente la información.



- **Iluminación teclado:** El teclado dispone de una luz de fondo para facilitar su uso en lugares de escasa visibilidad, por lo que hubo que controlar los valores que ofrecía el sensor de luminosidad para encender o apagar la luz según correspondiera.
- **Contraste:** El contraste del display es posible variarlo y hay que gestionar los valores del mismo según desea el usuario.
- **Iluminación pantalla:** El display también se puede iluminar para facilitar la lectura de los datos. Sin embargo, no siempre puede estar encendido pues deslumbra y además hay que evitar el desgaste del mismo y consumo que genera.
- **Información de pulsación:** El teclado al no ser mecánico necesita de un *feedback* para saber que se ha pulsado correctamente una tecla. Sin embargo este aviso, que es sonoro, solo se produce cuando la tecla pulsada tiene algún uso dentro del menú en el que se encuentre el usuario.
- **Optimización teclado:** Debido a la naturaleza del teclado, multitud de ocasiones ocurrían pulsaciones de teclas involuntariamente, porque hubo que optimizar la captura de estas teclas para que no interfirieran en el correcto funcionamiento de la aplicación.
- **Gestionar Wi-Fi:** Para el mantenimiento del sistema, se puede conectar remotamente con la máquina mediante Wi-Fi, por lo que hubo crear un script de activación y desactivación para controlarla, mediante una puerta trasera basada en una combinación de teclas.

## 5. Requisitos

En esta sección se definen los requisitos de software específicos de la aplicación. Para ello se usa un modelo de tabla como el siguiente:

Código	RF_YYY-ZZ
Título	
Prioridad	
Necesidad	
Verificabilidad	
Estabilidad	
Descripción	

Tabla 1: Tabla tipo para los requisitos

La plantilla de requisitos consta de los siguientes campos:

- Identificador: RX\_YYY-ZZ, donde X será F (funcional) o N (no funcional). YYY es el tipo de requisito, y ZZ una cifra que identifica unívocamente a cada requisito.  
Ej.: RF\_FUN-01 (Requisito funcional número 1).  
RN\_INT-02 (Requisito no funcional de interfaz de usuario número 2).

Los requisitos no funcionales son los siguientes:

- Interfaz (INT) Ej. RN\_INT-01.
- Interfaces externas (IEX) Ej. RN\_IEX-01.
- Prestaciones (PRE) Ej. RN\_PRE-01.
- Escalabilidad (ESC) Ej. RN\_ESC-01.
- Título: nombre del requisito que lo identifica unívocamente entre todos los demás.
- Prioridad: es necesario asignar una prioridad a cada requisito para facilitar una planificación del desarrollo lo más correcta posible. Las medidas de prioridad son alta, media y baja.
- Necesidad: los requisitos que sean imprescindibles para el usuario no podrán ser eliminados, mientras que el resto podrán ser opcionales, pudiendo ser suprimidos siempre y cuando haya algún motivo de peso que justifique esta supresión. Los valores que ha de tomar este atributo serán, en orden de necesidad creciente: opcional, deseable y esencial.
- Verificabilidad: este atributo indica si es posible comprobar si el sistema aplica el requisito, es decir, si se ha incluido en el diseño. Las medidas para este





campo son alta (en cuyo caso el requisito será fácilmente comprobable que ha sido introducido en el diseño), media y baja.

- Estabilidad: a lo largo del proceso de desarrollo hay requisitos que permanecen estables permanentemente, mientras que otros son inestables y están sujetos a posibles modificaciones. Por ello, es necesario especificar esta circunstancia mediante la inclusión de esta cualidad. Existirán dos tipos de valores para este campo: estable (el requisito no sufrirá modificaciones) e inestable (el requisito podrá ser modificado durante el proceso de desarrollo).
- Descripción: objetivos que describen el requisito y que el sistema debe cumplir.

### 5.1. Funcionales

Los requisitos funcionales definen las acciones que se pueden llevar a cabo en la aplicación.

A continuación se presentan dichas acciones en forma de requisitos.

<b>Código</b>	<b>RF_FUN-01</b>
<b>Título</b>	Indicador de teléfono.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	El indicador del teléfono ha de encenderse cuando éste se descuelga. Al colgarse, deberá apagarse este led. Mientras esté activado el manos libres deberá parpadear.

Tabla 2: Requisito RF\_FUN-01

<b>Código</b>	<b>RF_FUN-02</b>
<b>Título</b>	Indicador de megafonía.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	El indicador de megafonía deberá activarse cuando la radio, tanto la VHF como la TETRA, se encuentre en modo megafonía cabina, o en megafonía público o en megafonía timbre. En los demás modos permanecerá apagado.

Tabla 3: Requisito RF\_FUN-02



<b>Código</b>	<b>RF_FUN-03</b>
<b>Título</b>	Indicador de interconexión.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	El indicador de interconexión deberá activarse cuando la radio, tanto la VHF como la TETRA, se encuentre en modo interconexión. En los demás modos permanecerá apagado.

Tabla 4: Requisito RF\_FUN-03

<b>Código</b>	<b>RF_FUN-04</b>
<b>Título</b>	Indicador de recepción.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	El indicador de recepción se encenderá en VHF cuando se reciba cualquier señal portadora por el canal en el que se encuentre la radio. Cuando acabe la portadora, el led permanecerá apagado. El indicador de recepción se encenderá en TETRA cuando se reciba transmisión por una llamada privada o a través del grupo en el que se esté conectado. Cuando acabe la transmisión, el led permanecerá apagado.

Tabla 5: Requisito RF\_FUN-04

<b>Código</b>	<b>RF_FUN-05</b>
<b>Título</b>	Indicador de transmisión.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	El indicador de transmisión se encenderá siempre que se pulse el PTT, ya sea el del microteléfono o el automático del sistema de manos libres.

Tabla 6: Requisito RF\_FUN-05



<b>Código</b>	<b>RF_FUN-06</b>
<b>Título</b>	Indicador “On / Off”.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	El led estará de color naranja parpadeante durante la carga del sistema. Cuando la consola esté apagada el led deberá estar de color rojo. Cuando la consola esté encendida el led deberá estar de color verde.

Tabla 7: Requisito RF\_FUN-06

<b>Código</b>	<b>RF_FUN-07</b>
<b>Título</b>	Encendido de la consola.
<b>Prioridad</b>	Alta
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	La consola siempre estará encendida cuando el <i>inverter</i> esté activado. Cuando este desactivado, se deberá encender mediante una pulsación prolongada de la tecla On / Off.

Tabla 8: Requisito RF\_FUN-07

<b>Código</b>	<b>RF_FUN-08</b>
<b>Título</b>	Apagado de la consola.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	La consola únicamente podrá apagarse cuando el <i>inverter</i> esté desactivado, mediante una pulsación prolongada de la tecla On / Off. Además, deberá apagarse automáticamente tras cinco minutos de inactividad.

Tabla 9: Requisito RF\_FUN-08



Código	RF_FUN-09
Título	Aviso acústico del teclado.
Prioridad	Media.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	Cada pulsación de una tecla que tenga asignada funcionalidad en el momento concreto de la pulsación, deberá emitir un pitido a modo de <i>feedback</i> .

Tabla 10: Requisito RF\_FUN-09

Código	RF_FUN-10
Título	Manos libres.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El manos libres deberá activarse mediante una pulsación prolongada de la tecla asignada a tal efecto. Se desactivará automáticamente al descolgar el microteléfono o mediante pulsación prolongada de la tecla asignada a tal efecto.

Tabla 11: Requisito RF\_FUN-10

Código	RF_FUN-11
Título	Regulación de volumen.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	Cada modo de radio y cada estado, microteléfono o manos libres, tendrá asociado un volumen, que deberá poder modificarse entre unos márgenes mediante seis niveles.

Tabla 12: Requisito RF\_FUN-11



<b>Código</b>	<b>RF_FUN-12</b>
<b>Título</b>	Regulación de contraste.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	El display tiene un contraste que deberá poder modificarse entre unos márgenes mediante seis niveles.

Tabla 13: Requisito RF\_FUN-12

<b>Código</b>	<b>RF_FUN-13</b>
<b>Título</b>	Control de la luz ambiental.
<b>Prioridad</b>	Baja.
<b>Necesidad</b>	Deseable.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se debería controlar la luminosidad del ambiente, y en función de ésta, encender o apagar la luz del teclado.

Tabla 14: Requisito RF\_FUN-13

<b>Código</b>	<b>RF_FUN-14</b>
<b>Título</b>	Luz del display.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	La luz del display deberá encenderse cada vez que se produzca una notificación o cuando el usuario pulse una tecla. La luz deberá desactivarse transcurridos 30 segundos de inactividad.

Tabla 15: Requisito RF\_FUN-14



Código	RF_FUN-15
Título	Mostrar pantalla principal.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	Si se está en una pantalla que no sea la principal, transcurridos 60 segundos de inactividad se deberá volver a mostrar la pantalla principal.

Tabla 16: Requisito RF\_FUN-15

Código	RF_FUN-16
Título	Pasar a radio VHF.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El sistema deberá pasar a radio VHF cuando el usuario así lo desee y siempre que esta radio se encuentre disponible. Además se deberá pasar a esta radio automáticamente cuando el sistema detecte avería en radio la TETRA.

Tabla 17: Requisito RF\_FUN-16

Código	RF_FUN-17
Título	Pasar a radio TETRA.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El sistema deberá pasar a radio TETRA cuando el usuario así lo desee y siempre que esta radio se encuentre disponible. Para ello habrá de introducirse un grupo válido en el que el sistema pueda registrarse.

Tabla 18: Requisito RF\_FUN-17



<b>Código</b>	<b>RF_FUN-18</b>
<b>Título</b>	Cambiar de canal en VHF.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	La radio VHF deberá cambiar de canal cada vez que el usuario pulse las teclas asociadas al efecto. Cuando se alcancen los márgenes del rango de canales se preguntará al usuario si se desea pasar a radio TETRA.

Tabla 19: Requisito RF\_FUN-18

<b>Código</b>	<b>RF_FUN-19</b>
<b>Título</b>	Cambiar de grupo en TETRA.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	El usuario podrá establecer el grupo al que asociarse con la radio TETRA. Si se introduce un grupo incorrecto se hará caso omiso y se mantendrá el grupo actual.

Tabla 20: Requisito RF\_FUN-19

<b>Código</b>	<b>RF_FUN-20</b>
<b>Título</b>	Pasar a modo megafonía cabina.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	La radio deberá pasar a megafonía cabina cuando el usuario pulse la tecla asociada a tal efecto.

Tabla 21: Requisito RF\_FUN-20



Código	RF_FUN-21
Título	Pasar a modo megafonía público.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	La radio deberá pasar a megafonía público cuando el usuario pulse la tecla asociada a tal efecto.

Tabla 22: Requisito RF\_FUN-21

Código	RF_FUN-22
Título	Pasar a modo megafonía timbre.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	La radio deberá pasar a megafonía timbre automáticamente cuando se haya activado el tirador de alarma. Únicamente se podrá pasar a este modo cuando el usuario pulse la tecla asociada a tal efecto si el tirador de alarma ya está activado. Una vez desactivado el tirador, se deberá volver al modo que hubiera previamente al accionamiento de la alarma.

Tabla 23: Requisito RF\_FUN-22

Código	RF_FUN-23
Título	Pasar a modo radiotelefonía.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	La radio deberá pasar a radiotelefonía cuando el usuario pulse la tecla asociada a tal efecto. Este será el modo de inicio por defecto.

Tabla 24: Requisito RF\_FUN-23





Código	RF_FUN-24
Título	Pasar a modo interconexión.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	La radio deberá pasar a interconexión cuando el usuario pulse la tecla asociada a tal efecto.

Tabla 25: Requisito RF\_FUN-24

Código	RF_FUN-25
Título	Pasar a modo local.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	La radio deberá pasar a modo local únicamente cuando la radio activa sea la radio TETRA y el usuario haya pulsado la tecla asociada a tal efecto.

Tabla 26: Requisito RF\_FUN-25

Código	RF_FUN-26
Título	Pasar a modo local.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El modo local se desactivará automáticamente tras 30 segundos de inactividad o cuando el usuario cambie de modo por sí mismo. Además deberá recuperar el modo anterior existente si el cambio ha sido involuntario.

Tabla 27: Requisito RF\_FUN-26



Código	RF_FUN-27
Título	Introducción de los datos del conductor.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El usuario podrá cambiar los datos referentes al conductor introduciendo los nuevos datos a través del menú asociado a esta tarea.

Tabla 28: Requisito RF\_FUN-27

Código	RF_FUN-28
Título	Introducción de los datos del tren.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El usuario podrá cambiar los datos referentes al tren introduciendo los nuevos datos a través del menú asociado a esta tarea.

Tabla 29: Requisito RF\_FUN-28

Código	RF_FUN-29
Título	Introducción de los datos de la línea.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El usuario podrá cambiar los datos referentes a la línea introduciendo los nuevos datos a través del menú asociado a esta tarea.

Tabla 30: Requisito RF\_FUN-29



<b>Código</b>	<b>RF_FUN-30</b>
<b>Título</b>	Recibir mensajes.
<b>Prioridad</b>	Media.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Media.
<b>Descripción</b>	La aplicación deberá poder recibir hasta un máximo de cinco mensajes que el usuario podrá leer a través del menú asociado a esta tarea. Los mensajes únicamente serán recibidos y podrán consultarse si la radio activa es la TETRA. Se mostrará un icono informativo en la pantalla principal cuando existan mensajes sin leer.

Tabla 31: Requisito RF\_FUN-30

<b>Código</b>	<b>RF_FUN-31</b>
<b>Título</b>	Borrar mensajes.
<b>Prioridad</b>	Media.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	El usuario podrá borrar los mensajes individualmente una vez leídos o bien borrar todos los guardados a través del menú asociado a esta tarea.

Tabla 32: Requisito RF\_FUN-31

<b>Código</b>	<b>RF_FUN-32</b>
<b>Título</b>	Recibir llamada privada.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	El usuario podrá recibir llamadas privadas estando en la radio TETRA. Se visualizará el número del llamante y se mostrará un icono especial para esta situación.

Tabla 33: Requisito RF\_FUN-32



Código	RF_FUN-33
Título	Enviar mensaje “Aviso a jefe de circulación”.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El usuario podrá mandar el mensaje pulsando la tecla asociada a tal efecto. Se mandará tras realizar una confirmación para el envío, la cual podrá aceptar o cancelar.

Tabla 34: Requisito RF\_FUN-33

Código	RF_FUN-34
Título	Enviar mensaje “Rebase punto de parada”.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El usuario podrá mandar el mensaje pulsando la tecla asociada a tal efecto. Se mandará tras realizar una confirmación para el envío, la cual podrá aceptar o cancelar.

Tabla 35: Requisito RF\_FUN-34

Código	RF_FUN-35
Título	Enviar mensaje “Señal en rojo”.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El usuario podrá mandar el mensaje pulsando la tecla asociada a tal efecto. Se mandará tras realizar una confirmación para el envío, la cual podrá aceptar o cancelar.

Tabla 36: Requisito RF\_FUN-35



Código	RF_FUN-36
Título	Enviar mensaje “Quedan viajeros en andén”.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El usuario podrá mandar el mensaje pulsando la tecla asociada a tal efecto. Se mandará tras realizar una confirmación para el envío, la cual podrá aceptar o cancelar.

Tabla 37: Requisito RF\_FUN-36

Código	RF_FUN-37
Título	Enviar mensaje “Abandono cabina”.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El usuario podrá mandar el mensaje pulsando la tecla asociada a tal efecto. Se mandará tras realizar una confirmación para el envío, la cual podrá aceptar o cancelar.

Tabla 38: Requisito RF\_FUN-37

Código	RF_FUN-38
Título	Enviar mensaje “Puertas abiertas”.
Prioridad	Alta.
Necesidad	Esencial.
Verificabilidad	Alta.
Estabilidad	Alta.
Descripción	El usuario podrá mandar el mensaje pulsando la tecla asociada a tal efecto. Se mandará tras realizar una confirmación para el envío, la cual podrá aceptar o cancelar.

Tabla 39: Requisito RF\_FUN-38



<b>Código</b>	<b>RF_FUN-39</b>
<b>Título</b>	Mensaje informativo.
<b>Prioridad</b>	Baja.
<b>Necesidad</b>	Opcional.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Media.
<b>Descripción</b>	Posibilidad de recibir ciertos mensajes que envíe el RackDaemon sobre el sistema, para ser mostrada al usuario durante un periodo limitado de tiempo y con iconos informativos en función de la importancia o tipo del mensaje.

Tabla 40: Requisito RF\_FUN-39

<b>Código</b>	<b>RF_FUN-40</b>
<b>Título</b>	Mostrar la hora.
<b>Prioridad</b>	Baja.
<b>Necesidad</b>	Opcional.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se podrá mostrar la hora actual del sistema en la pantalla principal de la aplicación.

Tabla 41: Requisito RF\_FUN-40

<b>Código</b>	<b>RF_FUN-41</b>
<b>Título</b>	Mostrar la cobertura.
<b>Prioridad</b>	Baja.
<b>Necesidad</b>	Opcional.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se podrá mostrar la cobertura de las radios mediante un icono representativo en la pantalla principal de la aplicación.

Tabla 42: Requisito RF\_FUN-41



<b>Código</b>	<b>RF_FUN-42</b>
<b>Título</b>	Encender Wi-Fi.
<b>Prioridad</b>	Media.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá poder encender el sistema Wi-Fi del tren mediante una combinación oculta de teclas. Se deberá oír un triple pitido en señal de la correcta activación

Tabla 43: Requisito RF\_FUN-42

<b>Código</b>	<b>RF_FUN-43</b>
<b>Título</b>	Apagar Wi-Fi.
<b>Prioridad</b>	Media.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá poder apagar el sistema Wi-Fi del tren mediante una combinación oculta de teclas. Se deberá oír un único pitido en señal de la correcta desactivación.

Tabla 44: Requisito RF\_FUN-43

<b>Código</b>	<b>RF_FUN-44</b>
<b>Título</b>	Mostrar información del sistema.
<b>Prioridad</b>	Baja.
<b>Necesidad</b>	Opcional.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se podrá mostrar información referente a las versiones software e información del hardware del equipo mediante una combinación oculta de teclas. Únicamente podrá realizarse cuando el inverter esté desactivado.

Tabla 45: Requisito RF\_FUN-44

## 5.2. No funcionales

Los requisitos no funcionales son otros requisitos que no forman parte de la funcionalidad principal de la aplicación. Son restricciones que se aplican, prestaciones necesarias de la aplicación, calidad y otros. Los requisitos no funcionales de este proyecto son los siguientes:

- **Requisitos de interfaz de usuario**

<b>Código</b>	<b>RN_INT-01</b>
<b>Título</b>	Menús.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Media.
<b>Descripción</b>	Los menús a mostrar deben ser iguales que los mostrados en el <b>¡Error! No se encuentra el origen de la referencia..</b>

Tabla 46: Requisito RN\_INT-01

<b>Código</b>	<b>RN_INT-02</b>
<b>Título</b>	Iconos gráficos.
<b>Prioridad</b>	Media.
<b>Necesidad</b>	Deseable.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Dado que se dispone de un display gráfico es aconsejable adjuntar mejoras gráficas al diseño de los menús en todos los apartados que sea posible.

Tabla 47: Requisito RN\_INT-02

<b>Código</b>	<b>RN_INT-03</b>
<b>Título</b>	Ayuda contextual.
<b>Prioridad</b>	Media.
<b>Necesidad</b>	Deseable.
<b>Verificabilidad</b>	Alta.





<b>Estabilidad</b>	Alta.
<b>Descripción</b>	En los menús en los que sea posible introducir una línea, sería aconsejable incluir la primera línea como título del menú, separada del resto por un delimitador.

Tabla 48: Requisito RN\_INT-03

<b>Código</b>	<b>RN_INT-04</b>
<b>Título</b>	Título de menú.
<b>Prioridad</b>	Media.
<b>Necesidad</b>	Deseable.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	En los menús en los que sea posible introducir una línea, sería aconsejable incluir la última línea como ayuda contextual, separada del resto por un delimitador.

Tabla 49: Requisito RN\_INT-04

- **Requisitos de interfaces externas**

<b>Código</b>	<b>RN_IEX-01</b>
<b>Título</b>	Interfaz con RackDaemon.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Alta.
<b>Estabilidad</b>	Media.
<b>Descripción</b>	El desarrollo debe englobarse en el flujo de ejecución de la librería GLib para adaptarse a la comunicación mediante D-Bus con el extremo que representan las interfaces del RackDaemon.

Tabla 50: Requisito RN\_IEX-01

- **Requisitos de prestaciones**

<b>Código</b>	<b>RN_PRE-01</b>
<b>Título</b>	Sistema embebido
<b>Prioridad</b>	Media.
<b>Necesidad</b>	Esencial.
<b>Verificabilidad</b>	Media.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Debido a que el objetivo del sistema es instalarse en un entorno embebido es necesario que se cuide al máximo la eficiencia de



	los algoritmos para evitar que la entrada interactiva sea lenta. Por lo tanto se minimizarán los datos a ser transferidos, la carga de la CPU y el uso de memoria.
--	--

Tabla 51: Requisito RN\_PRE-01

- **Requisitos de escalabilidad**

<b>Código</b>	<b>RN_ESC-01</b>
<b>Título</b>	Ampliación de menús.
<b>Prioridad</b>	Media.
<b>Necesidad</b>	Opcional.
<b>Verificabilidad</b>	Media.
<b>Estabilidad</b>	Media.
<b>Descripción</b>	Se debe cuidar la posibilidad de que la especificación de los menús se amplíe en un futuro, por lo que la modificación de los mismos no ha de ser costosa.

Tabla 52: Requisito RN\_ESC-01

## **6. Implantación**

Una vez realizado el sistema se procedió a su integración e instalación. Como es lógico durante este proceso surgieron diferentes problemas que pasan a explicarse a continuación.

### **6.1. Problemas de implantación**

Una vez testado los diversos programas, había que integrarlos e implantarlos en el sistema real.

El principal problema al que nos enfrentamos fue el del rendimiento del sistema. Los simuladores en los que se probaron los programas tenía una potencia mucho mayor que la que nos encontramos en el sistema final.

En un primer momento las pruebas parecían ser satisfactorias, pero en el momento en que se le empezaba a dar una ligera carga al sistema el rendimiento decrecía mucho debido a multitud de factores que hubo que ir estudiando detenidamente.

El primer problema que nos dimos cuenta es que los logs del sistema eran demasiado complejos y escriban demasiadas trazas constantemente debido a la multitud de información proveniente de las radios, por lo que hubo que limitarlo, no solo por la carga que se le daba a la CPU sino también porque afectaría a la vida útil del disco que llevaban las máquinas, que eran realmente tarjetas de memoria. Estas tarjetas tienen un número máximo de escrituras posibles sobre ellas y el sistema había de ser lo suficientemente duradero como para evitar tener que remplazar estas tarjetas constantemente.

Otro problema que afectó al rendimiento del sistema fue la gestión del display. Una mejora realmente sustancial respecto al antiguo sistema fue la introducción de un display gráfico que nos ofrecía multitud de posibilidades para mostrar la información a los maquinistas. Por tanto se decidió crear transiciones entre los menús, iconos de información y demás mejoras con el fin de sorprender al cliente con nuestro producto. No obstante, estas mejoras gráficas, que en nuestros simuladores funcionaban a la perfección, vimos que en el sistema real le daban mucha carga a la CPU. Pero había un problema también muy destacable y es que finalmente esta mejora parecía contraproducente, debido a que el refresco de la pantalla era realmente lento, y lo que en un principio era para realzar las propiedades del display, realmente hacían lo contrario. No obstante, como se verá más tarde, obviamente no se descartó usar estas mejoras, porque realmente era un punto muy positivo dentro de nuestra oferta.



El display nos dio un problema más. Al arrancar el sistema, tras una pantalla informativa que indicaba al usuario que el sistema estaba cargando, el display comenzaba a comportarse de manera errática durante unos segundos, mostrando figuras que podrían asemejarse a códigos de barras en la pantalla. El problema se incrementaba cuando el dispositivo de inicio del tren, el inverter, se encontraba desconectado, puesto que esta figura se quedaba instaurada en el display hasta el encendido del mismo.

Esto no era un gran problema, pues realmente no afectaba en nada la funcionalidad del sistema, pues una vez encendido por primera vez el display, éste pasaba a funcionar de manera correcta, pero sin embargo, era un problema en la presentación del mismo que no nos gustaba de cara a satisfacer completamente al cliente, principalmente cuando el tren estaba apagado, pues se quedaba permanentemente el display de esta manera.

Este problema nos resulto realmente complicado solucionarlo, debido principalmente a que no sabíamos la causa del mismo y por lo tanto tampoco sabíamos por donde buscar una solución al mismo. Finalmente, y tras estar a punto de dejarlo estar, puesto que la mayoría de las veces, cuando arranca el sistema, los trenes están encendidos y este bug casi pasaría desapercibido, conseguimos dar con la solución.

Sin embargo, el mayor problema que tuvo nuestro sistema fue el teclado.

Debido a las referencias que teníamos de antiguos sistemas, y de la insistencia del cliente de que fuera un sistema duradero y de alguna manera, irrompible, se decidió insertar este tipo de teclado en nuestra oferta. Con él conseguiríamos un doble efecto, tener un teclado duradero y además un sistema muy atractivo de cara al usuario final.

No obstante, una vez que empezamos a realizar pruebas con él, y sobre todo gracias a la experiencia previa que teníamos sobre como usaban el teclado los maquinistas, vimos que el teclado no era efectivo. Esto era debido a la sensibilidad del mismo.

Comprobamos como para una sola pulsación, muchas veces se pulsaban involuntariamente sus teclas cercanas.

Esto en muchos casos pasaba desapercibido, puesto que por requerimientos del cliente, las teclas que no tenía funcionalidad en un cierto menú quedaban desactivadas, pero sin embargo en la pantalla principal que la mayoría de ellas sí que



están activas, si que supone un problema. También era bastante problemático cuando se insertaban datos, ya que muchas veces se tecleaban números no marcados y al darle a la tecla borrar, que está al lado de la de aceptar, se validaban datos no deseados. Esto incluso ocurría incluso al colgar el microteléfono.

El problema de la sensibilidad del teclado también afectaba al hecho de que había teclas que requerían una pulsación prolongada para activarse, como es el caso de la tecla de encendido y apagado y a la del manos libres.

Finalmente, tras muchos cambios e intentos de mejorar los drivers y la aplicación que gestionaba la consola, se llegó a una solución altamente óptima.

Por último, también hubo muchos cambios por parte del cliente según iba viendo maquetas del producto. Tras mucho tiempo empleado en implementar ciertas cosas que el cliente inicialmente deseaba, al verlas implantadas en el sistema real no estaba satisfecho con el resultado final y algunas opciones las desechaba completamente y otras exigía un cambio bastante considerable.

Estos problemas normalmente eran sugeridos por los usuarios finales, puesto que algunas de las mejoras introducidas en este nuevo producto afectaban a la funcionalidad que era ligeramente diferente a las existentes en la solución anterior y a pesar de poder tener más funcionalidad optaban por la más simple o parecida a la que ya conocían.

## **6.2. Soluciones aportadas**

Como se ha visto anteriormente, y como era de esperar, surgieron varios problemas a la hora de realizar la integración e implantación del sistema, por lo que se tuvieron que estudiar diferentes soluciones para solventar estos imprevistos. Aquí veremos las soluciones referentes a los problemas que afectaron al programa de gestión de la consola.

Primeramente, como se vio anteriormente nos encontramos con cantidad de información volcada sobre los logs. Esto era en parte producto del exceso de datos que nos transferían las radios, por lo que tuvimos que limitar el refresco de información para no saturar la CPU. Además de esto también pudimos comprobar que enviábamos cierta información redundante que procedimos a eliminar.

Sin embargo, los logs del sistema eran requisitos necesarios por parte del cliente, y también útiles en nuestro proyecto para poder depurarlo correctamente, por lo que creamos dos entornos de ejecución, uno en modo pruebas y otro en modo



depuración. Esto básicamente se consiguió pasando un parámetro para la ejecución del programa, y en función de este parámetro se generarían distintos logs, uno más completo para el modo depuración, y otro con información concreta y específica sobre el estado del sistema para el modo de ejecución normal.

Como también se explico previamente, otro gran problema de rendimiento venía dado por el refresco del display.

Inicialmente, la aplicación se visualizaba correctamente. Sin embargo, el momento crítico y cuando peor se comportaba la aplicación era en los refrescos de la pantalla principal. Esto era debido a la multitud de iconos que se mostraban, además de las líneas de texto que estaban escritas al completo.

Además de esto, también se apreciaba este problema alguna vez en las transiciones de menús.

Inicialmente, el display poseía un driver que hacía que en vez de refrescar la pantalla entera, solo se modificaran los píxeles que había cambiado de estado. Se pensó que esta solución era suficiente para no darle carga a la CPU. Sin embargo, inicialmente, la aplicación no iba a mostrar tantos iconos y figuras en la pantalla, que eran lo que afectaba realmente al rendimiento.

Debido a esta decisión, inicialmente la aplicación lo que hacía, por tanto, era modificar únicamente las partes que se veían afectadas. El problema que surgía era que la pantalla se refrescaba secuencialmente, pues la CPU no era lo suficientemente rápida, y creaba entonces un efecto muy inadecuado.

La primera solución por la que se optó fue por dividir el display por líneas y modificar la línea entera de una vez, para que no se apreciara el cambio dentro de ella. Sin embargo, esta solución era efectiva principalmente en toda la aplicación, excepto cuando se mostraba la pantalla principal.

Con esta primera solución ya prácticamente se solucionaban todos los problemas, excepto cuando se volvía a mostrar la pantalla principal proviniendo desde cualquier otro menú. Con esta solución la carga de la CPU era menor, pues aunque generaba más imágenes, se plasmaban menos cambios en el display por lo que la mejora venía dada por ese aspecto.

Por lo tanto, viendo como se comportaba realmente la aplicación y los drivers del display, optamos finalmente por generar cada vez que se iba a modificar el display, una imagen completa del mismo, que sería la única que se plasmara en el mismo.



Esta solución generaba una mayor complejidad del algoritmo para mostrar la pantalla, así como un mayor tráfico de datos a través del D-Bus.

Sin embargo, el tráfico de datos por el D-Bus estaba muy limitado previamente y por este lado no hubo inconvenientes. Además de esto, se observó como la CPU podía generar sin problemas la imagen completa, puesto que aunque la imagen era de mayor resolución y el algoritmo era más complejo, lo que realmente cargaba la CPU era volcarlo sobre el display.

Con esta solución ya no se observaba ningún problema durante los refrescos, pues sólo se plasmaba una imagen de una vez. Sin embargo, esta solución hizo que la carga de la CPU durante las transiciones de los menús y en los valores indicativos que parpadean tuviera un pico casi del 100%. No obstante, esto se produce solamente durante unas pocas décimas de segundo, por lo que se dio como válida finalmente esta solución.

Además de esto, el display nos dio un problema más como se comentó previamente, y era el hecho de que iniciándose la aplicación se mostraban ciertas figuras que permanecían fijas hasta que se encendía la consola.

Este problema fue bastante complejo debido a que no sabía la causa del mismo. No afectaba a la funcionalidad de la consola, pero por el contrario sí que ofrecía una mala impresión al cliente.

El problema venía dado por la pantalla de inicio del sistema, pues sin cargarla este efecto no se producía. En un primer momento se pensó que el problema viniera del primer programa que cargaba esta pantalla de inicio, pues era independiente de la aplicación de gestión de la consola. Sin embargo comprobamos que controlaba correctamente el apagado del display una vez finalizada su ejecución.

Descartado que esa fuera la raíz del problema, se pensó en el contraste, pues había dado ciertos problemas, por lo que se procedió a reducirse completamente al display pero tampoco solucionaba el problema.

Más tarde se concluyó que el problema venía dado por la lentitud con la que se instanciaba un objeto que controlaba el display a través de los driver específicos. El problema era por tanto, que se daban órdenes al display antes de que éste estuviera preparado para recibirlas. Para ello se hizo un *sleep* prudencial al inicio, que no retrasaba el inicio de la consola de una forma sustancial, pero que permitía sin embargo que se creara completamente el objeto asociado al display.



No obstante, aunque esto solucionaba en parte el problema no acabó de resolverse completamente, por lo que finalmente se optó por generar un objeto inicial de la clase que controlaba el display, para que se encargara de encender el display para inmediatamente apagarlo.

Con esta solución se consiguió definitivamente quitar las imágenes extrañas que se producían al inicio de la aplicación.

Por otro lado, con el problema con el teclado se tardó bastante en encontrar una solución realmente efectiva.

El problema estaba claro, se efectuaban pulsaciones involuntarias, por lo que habría que evitarlas de la mejor manera posible. Buscando soluciones, se fueron aplicando consecutivamente, una sobre otra, que iban perfeccionando la respuesta que ofrecía, hasta que finalmente se encontró la solución definitiva.

Primeramente, por una mezcla de problemática hardware y software, había teclas que se bloqueaban. Por este motivo, el driver que controlaba el teclado dejó de ofrecer la repetición de una tecla que se mantuviera pulsada. Esta solución evitó que se bloqueara el teclado, pero por el contrario dificultó el algoritmo empleado para controlar diversas teclas pulsadas a la vez (diversas combinaciones de teclas para las puertas traseras) y controlar el tiempo que se mantenía una tecla pulsada (por ejemplo, la tecla de encendido) para que hiciera su función.

Más tarde, y comprobando que esta solución no era del todo efectiva, se paso a comprobar que una tecla estuviera pulsada un cierto tiempo para activarse, al estilo de la tecla de encendido, puesto que las pulsaciones involuntarias duraban escasos milisegundos.

Esta solución primeramente se creó en la aplicación de gestión de la consola, y más tarde también se implementó en el driver. Sin embargo, no se dio el visto bueno final hasta que además de esta comprobación, se impidiera la activación de una segunda tecla activada inmediatamente después de otra.

Con esta serie de comprobaciones finalmente se obtuvo un teclado realmente eficaz, sin pulsaciones involuntarias y con un mínimo tiempo de pulsado para activar la función de cada tecla.

Finalmente, también se tuvo que dar cabida a todos los cambios que solicitaba el cliente a medida que se iba implantando la solución.





### **6.3. Comentarios basados en la experiencia**

Tras este periodo de implantación, la conclusión más importante a la que llego es que es fundamental tener en alta consideración la multitud de problemas e inconvenientes que pueden surgir, por lo que hace que sea de vital importancia tener una buena planificación para hacer frente a estos imprevistos.

Siempre surgirán imprevistos que no han sido planificados previamente, por lo que es de vital importancia ir cumpliendo los plazos para poder asegurar la finalización del proyecto a tiempo. En este caso en particular, los mayores problemas vinieron dados por el display y el teclado de la consola, pero gracias al sobre esfuerzo del equipo de trabajo se consiguieron resolver las incidencias que se presentaron.

También he podido comprobar lo importante que es tener una buena comunicación al cliente, ya que se introdujeron varias mejoras que estimábamos útiles y que mejoraban la solución considerablemente, pero que finalmente fueron desechadas una vez desarrolladas, pues el cliente no estaba conforme con ellas.

Pero no sólo la comunicación es importante, sino conocer al cliente en sí. Hay que adaptarse a sus necesidades, conocer lo que reclama y por supuesto su manera de trabajar, para que a la hora de desarrollar una solución a medida, el resultado sea lo más satisfactorio posible, pues el hecho de realizar un producto realmente atractivo para el cliente es vital para que en un futuro seamos nosotros, y no otros, los encargados de dar solución a sus problemas.

Por último, remarcar la importancia del equipo de trabajo, puesto que es muy importante que vaya sincronizado para ir cumplimiento los hitos propuestos en la planificación, puesto que en muchos casos, el no llegar a tiempo cierta parte del equipo de desarrollo es equivalente al fallo total del mismo. Es fundamental que haya un buen entorno de trabajo, que facilitará la comunicación entre los miembros del grupo, así como una mayor rapidez a la hora de encontrar soluciones a los diversos problemas que hay que afrontar. Y por supuesto, pensar en favorecer y ayudar al equipo y a los que trabajan contigo, desde las cosas más básicas, puesto que al final, nuestras metas son las mismas y se conseguirán mejores resultados.

## **7. Evaluación del sistema**

El propósito de esta sección es la de establecer la definición de las distintas pruebas, estrategia, criterios de aceptación, entorno y configuración informática para las mismas, así como detallar los casos de pruebas diseñados según la estrategia establecida con el objeto de verificar el correcto funcionamiento del sistema para las pruebas de las consolas embarcadas de Metro de Madrid.

### **7.1. Pruebas funcionales**

El plan de pruebas se definirá y ejecutará de acuerdo a metodologías de pruebas de software, buscando siempre un punto de equilibrio entre el esfuerzo necesario para llevar a cabo las pruebas y las necesidades concretas del proyecto. El objetivo de la ejecución del plan de pruebas no es comprobar que el diseño es infalible, sino que las decisiones arquitecturales tomadas hasta el momento son adecuadas y que permitirán que el desarrollo resultante del proyecto alcance los requisitos del sistema.

Las pruebas propuestas, por tanto, serán de caja negra, en las que el sistema a probar será el prototipo desarrollado. Con esto, las pruebas se definirán como casos de prueba, especificando las entradas a aportar al sistema junto con el resultado que se espera obtener de las mismas. Los casos de pruebas definidos tratan de comprobar el funcionamiento del sistema en aspectos clave, tales como las integraciones que éste debe soportar.

El objetivo de la fase de pruebas es el de someter al sistema de información a una serie de verificaciones encaminadas a garantizar el nivel de fiabilidad y calidad deseado.

La fase de Pruebas debe comprobar el sistema en su totalidad, haciendo énfasis en los siguientes aspectos:

- Verificar que funcionalmente se cumplen los requisitos recogidos en la fase de Análisis de Requisitos.
- Verificar que se han implementado correctamente todos los requisitos de diseño del sistema.
- Detectar cualquier error de codificación incurrido en la fase de implementación.
- Asegurar la adecuación a los estándares de análisis, diseño, codificación, nomenclatura, etcétera, existentes.

- Comprobar que el sistema es capaz de realizar las operaciones básicas con los volúmenes de información requeridos. El sistema será examinado bajo las condiciones más extremas.
- Asegurar la validación del sistema por parte de los usuarios.

## 7.2. Casos de prueba

El objetivo de este punto es plantear un protocolo de pruebas para que se pueda conseguir la máxima fiabilidad del software y del hardware suministrado y para comprobar que el sistema es capaz de cumplir con los requisitos impuestos tanto por Metro de Madrid como por los mantenedores y montadores.

En este documento no se incluyen las pruebas de funcionalidades no especificadas o de sistemas no contemplados, así como de algunos errores del sistema que no se pueden reproducir en el entorno de pruebas.

La tabla tipo usada para definir los casos de prueba será la siguiente.

Identificador	CP_XXX-YY	
Descripción		
Prerrequisitos		
Paso	Procedimiento	Resultado esperado

Tabla 53: Tabla tipo para los casos de prueba

- Identificador: Será del tipo CP\_XXX-YY donde XXX es el tipo de prueba funcional, y YY una cifra que identifica unívocamente a cada prueba.

Ej.: CP\_MEGAFO-01 (Caso de prueba de megafonía número 1).

CP\_TECLADO-01 (Caso de prueba de teclado número 1).

Los tipos de casos de prueba son los siguientes:

- Consola del conductor (CONSOLA) Ej. CP\_CONSOLA-01.
- Teclado de la consola (TECLADO) Ej. CP\_TECLADO-01.
- Display de la consola (DISPLAY) Ej. CP\_DISPLAY-01.
- Microteléfono y altavoz de la consola (MICROA) Ej. CP\_MICROA-01.
- Sistema de radiotelefonía (RADIOTEL) Ej. CP\_RADIOTEL-01.
- Sistema de megafonía (MEGAFO) Ej. CP\_MEGAFO-01.
- Mensajes de estado (MESTADO) Ej. CP\_MESTADO-01.
- Menús de consola (MENUCON) Ej. CP\_MENUCON-01.



- Descripción: Descripción del caso de prueba.
- Prerrequisitos: Entorno necesario previo para poder desarrollar la prueba correctamente.
- Paso: Orden en el que se realizaran las comprobaciones dentro del caso de prueba.
- Procedimiento: Descripción de la prueba que hay que llevar a cabo.
- Resultado esperado: Resultado óptimo esperable para el caso de prueba.

Los casos de prueba son los siguientes:

Identificador	CP_CONSOLA-01	
Descripción	Indicador Teléfono.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Descolgar el microteléfono.	Comprobar que el indicador “Teléfono” se enciende en color verde.
2	Colgar el microteléfono.	Comprobar que el indicador “Teléfono” se apaga.

Tabla 54: Caso de prueba CP\_CONSOLA-01

Identificador	CP_CONSOLA-02	
Descripción	Indicador “MGF”.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Modo radiotelefonía.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla de “Megafonía” (teclas 0, 1 o 2).	Comprobar que el indicador “MGF” se enciende en color rojo al activar la megafonía.
2	Pulsar la tecla de “Radiotelefonía” (tecla 5).	Comprobar que el indicador “MGF” se apaga al desactivar la megafonía.

Tabla 55: Caso de prueba CP\_CONSOLA-02



Identificador	CP_CONSOLA-03	
Descripción	Indicador “INT”.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Modo radiotelefonía.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla de “INT” (tecla 3).	Comprobar que el indicador “INT” se enciende en color azul al activar la intercomunicación y se apaga al desactivarla.
2	Pulsar la tecla de “Radiotelefonía” (tecla 5).	Comprobar que el indicador “INT” se apaga al desactivar la intercomunicación.

Tabla 56: Caso de prueba CP\_CONSOLA-03

Identificador	CP_CONSOLA-04	
Descripción	Indicador “Tx”.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Modo radiotelefonía.		
Paso	Procedimiento	Resultado esperado
1	Realizar una transmisión vía radio desde la consola (por ejemplo, descolgar teléfono, pulsar PTT y hablar).	Comprobar que el indicador “TX” se enciende en color rojo al transmitir datos vía radio. Nota: en caso de comunicación dúplex, se enciende también el indicador “RX”.
2	Dejar de transmitir vía radio desde la consola.	Comprobar que el indicador “TX” permanece apagado al no transmitir.

Tabla 57: Caso de prueba CP\_CONSOLA-04



Identificador	CP_CONSOLA-05	
Descripción	Indicador “Rx”.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Modo radiotelefonía.		
Paso	Procedimiento	Resultado esperado
1	Realizar una transmisión vía radio hacia la consola (por ejemplo, transmisión desde el puesto de mando).	Comprobar que el indicador “RX” se enciende en color amarillo al recibir datos vía radio. Nota: en caso de comunicación dúplex, se enciende también el indicador “TX”.
2	Dejar de transmitir vía radio hacia la consola.	Comprobar que el indicador “RX” permanece apagado al no recibir datos vía radio.

Tabla 58: Caso de prueba CP\_CONSOLA-05

Identificador	CP_CONSOLA-06	
Descripción	Indicador “On / Off” a off.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Enviar la señal de cabina activa desde el sistema de señales del tren (o desde el simulador) y pulsar el botón “On / Off” para apagar la consola.	Comprobar que la consola no se apaga y el indicador “On / Off” permanece encendido en color verde. De esta manera se asegura que el sistema de radiotelefonía siempre esté activo con el tren en marcha.
2	Enviar la señal de cabina no activa desde el sistema de señales del tren (o desde el simulador) y pulsar el botón “On / Off” para apagar la consola.	Comprobar que el indicador “On / Off” se enciende en color rojo y la consola se apaga.

Tabla 59: Caso de prueba CP\_CONSOLA-06



Identificador	CP_CONSOLA-07	
Descripción	Indicador “On / Off” a On.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Consola apagada.		
Paso	Procedimiento	Resultado esperado
1	Enviar la señal de cabina activa desde el sistema de señales del tren (o desde el simulador) y pulsar el botón “On / Off” para encender la consola.	Comprobar que el indicador “On / Off” se enciende en color verde y se activa la consola.
2	Enviar la señal de cabina no activa desde el sistema de señales del tren (o desde el simulador) y pulsar el botón “On / Off” para encender la consola.	Comprobar que la consola se enciende y el indicador “On / Off” permanece encendido en color verde.

Tabla 60: Caso de prueba CP\_CONSOLA-07

Identificador	CP_CONSOLA-08	
Descripción	Inicio correcto de los datos mostrados.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Consola en pantalla principal.		
Paso	Procedimiento	Resultado esperado
1	Comprobar los datos mostrados en la consola.	Los datos mostrados han de ser los almacenados en la base de datos.

Tabla 61: Caso de prueba CP\_CONSOLA-08

Identificador	CP_CONSOLA-09	
Descripción	Encendido automático de la consola.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Consola apagada y señal de cabina inactiva.		
Paso	Procedimiento	Resultado esperado
1	Enviar la señal de cabina activa desde el sistema de señales del tren (o desde el simulador.	Comprobar que la consola se enciende automáticamente al recibir la señal de cabina activa.

Tabla 62: Caso de prueba CP\_CONSOLA-09



Identificador	CP_CONSOLA-10	
Descripción	Apagado automático de la consola.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Señal de cabina activa.		
Paso	Procedimiento	Resultado esperado
1	Enviar la señal de cabina inactiva desde el sistema de señales del tren (o desde el simulador) y no realizar ninguna operación.	Comprobar que con la consola encendida, se apaga automáticamente al pasar cinco minutos desde la recepción de la señal de cabina inactiva.

Tabla 63: Caso de prueba CP\_CONSOLA-10

Identificador	CP_CONSOLA-11	
Descripción	Rendimiento y fiabilidad de la consola.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Señal de cabina activa.		
Paso	Procedimiento	Resultado esperado
1	Creación de un script de pruebas con una carga intensiva de todas las operaciones posibles, a partir del cual se realizarán pruebas de carga del sistema.	Una vez realizadas las pruebas de carga, comprobar que los logs que han sido generados no contengan ningún error del sistema y el procesador del equipo no se ha saturado ni bloqueado.

Tabla 64: CP\_CONSOLA-11

Identificador	CP_TECLADO-01	
Descripción	Comprobación del teclado.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Inspeccionar el teclado de la consola.	Comprobar que todas las teclas del teclado funcionan.

Tabla 65: Caso de prueba CP\_TECLADO-01





Identificador	CP_TECLADO-02	
Descripción	Aviso acústico del teclado.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Pulsar alguna tecla de la consola con funcionalidad activa.	Comprobar que se recibe un aviso acústico cuando se pulsa una tecla de la consola con funcionalidad activa.

Tabla 66: Caso de prueba CP\_TECLADO-02

Identificador	CP_DISPLAY-01	
Descripción	Inspección del display.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Inspeccionar el display LCD.	Comprobar que se encuentra encendido y con las cinco líneas activas.

Tabla 67: Caso de prueba CP\_DISPLAY-01

Identificador	CP_DISPLAY-02	
Descripción	Aumentar contraste.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Contraste” del teclado de la consola.	Comprobar que se accede al menú Contraste.
2	Pulsar la tecla “Subir”.	Comprobar que aumenta el contraste del display LCD (regulable en 6 niveles) y que se muestra en el display el nivel ajustado.

Tabla 68: Caso de prueba CP\_DISPLAY-02



Identificador	CP_DISPLAY-03	
Descripción	Disminuir contraste.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Contraste” del teclado de la consola.	Comprobar que se accede al menú Contraste.
2	Pulsar la tecla “Bajar”.	Comprobar que disminuye el contraste del display LCD (regulable en seis niveles) y que se muestra en el display el nivel ajustado.

Tabla 69: Caso de prueba CP\_DISPLAY-03

Identificador	CP_MICROA-01	
Descripción	Aumentar nivel de audio.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Volumen” del teclado.	Comprobar que se accede al menú Volumen.
2	Pulsar la tecla “Subir”.	Comprobar que aumenta el volumen del altavoz (regulable en seis niveles) y que se muestra en el display el nivel ajustado.

Tabla 70: Caso de prueba CP\_MICROA-01

Identificador	CP_MICROA-02	
Descripción	Disminuir nivel de audio.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Volumen” del teclado.	Comprobar que se accede al menú Volumen.
2	Pulsar la tecla “Bajar”.	Comprobar que disminuye el volumen del altavoz (regulable en



		seis niveles) y que se muestra en el display el nivel ajustado.
--	--	---

Tabla 71: Caso de prueba CP\_MICROA-02

Identificador	CP_MICROA-03	
Descripción	Volumen mínimo de audio.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Disminuir el nivel de volumen al mínimo en el altavoz.	Disminuir al mínimo el nivel del audio.
2	Realizar una transmisión de audio (simulando Puesto de Mando).	Comprobar que existe audio pese a estar el nivel de audio en el mínimo configurable, de manera que siempre se escuche audio.

Tabla 72: Caso de prueba CP\_MICROA-03

Identificador	CP_RADIOTEL-01	
Descripción	Selección de radio activa TETRA.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio VHF activa.		
Paso	Procedimiento	Resultado esperado
1	Activar el modo Radio Telefonía pulsando la tecla correspondiente del teclado de la consola.	Se activa el modo Radio Telefonía.
2	Pulsar la tecla “Menú” del teclado de la consola.	Aparecerá el menú de selección con las opciones:  1. Datos de circulación 2. Establecer radio 3. Buzón TETRA
3	Seleccionar la opción 2 del menú, pulsando las teclas “Subir”/“Bajar” hasta posicionarse sobre la opción y pulsando “Aceptar” o bien pulsando directamente “2”.	Se accede al sub-menú que permite seleccionar la radio activa en el sistema:  1. Radio VHF 2. Radio TETRA



4	Seleccionar la segunda opción del submenú y Aceptar.	Se muestra la pantalla de selección del grupo de línea o de depósito.
5	Elegir el grupo de línea o de depósito deseado y Aceptar.	<p>Comprobar que la radio se afilia al sistema y se muestra en el Display la información del grupo de línea seleccionado.</p> <p>Para grupo de línea: LXX NXX Ag XXXXX Grupo línea XX Radiotelefonía</p> <p>Para grupo de depósito: LXX NXX Ag XXXXX Grupo dpto = XX Radiotelefonía</p>

Tabla 73: Caso de prueba CP\_RADIOTEL-01

Identificador	CP_RADIOTEL-02	
Descripción	Cambio de grupo en TETRA.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio TETRA activa		
Paso	Procedimiento	Resultado esperado
1	Pulsar las teclas “Canal +” o “Canal -” del teclado de la consola para seleccionar el canal deseado.	Se muestra la pantalla de selección del grupo de línea o de depósito.
2	Elegir el grupo de línea o de depósito deseado y Aceptar.	Comprobar que la radio se afilia al sistema y se muestra en el Display la información del grupo de línea seleccionado:  Para grupo de línea: LXX NXX Ag XXXXX Grupo línea XX Radiotelefonía  Para grupo de depósito: LXX NXX Ag XXXXX Grupo dpto = XX Radiotelefonía

Tabla 74: Caso de prueba CP\_RADIOTEL-02



Identificador	CP_RADIOTEL-03		
Descripción	Selección de radio activa VHF.		
Prerrequisitos			
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio TETRA activa.			
Paso	Procedimiento	Resultado esperado	
1	Pulsar la tecla “Menú” del teclado de la consola.	Aparecerá el menú de selección con las opciones:  1. Datos de circulación 2. Establecer radio 3. Buzón TETRA	
2	Seleccionar la opción 2 del menú, pulsando las teclas “Subir”/“Bajar” hasta posicionarse sobre la opción y pulsando “Aceptar” o bien pulsando directamente “2”.	Se accede al sub-menú que permite seleccionar la radio activa en el sistema:  1. Radio VHF 2. Radio TETRA	
3	Seleccionar la opción 1 del submenú y Aceptar.	Comprobar que se activa la radio VHF y se muestra en el display la información del canal y grupo. seleccionado: VHF LINEA X TREN XX CANAL=X LIN=XX,XX,XX Radiotelefonía	

Tabla 75: Caso de prueba CP\_RADIOTEL-03

Identificador	CP_RADIOTEL-04	
Descripción	Cambio de canal en VHF.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio VHF activa.		
Paso	Procedimiento	Resultado esperado
1	Pulsar las teclas “Canal +” o “Canal -” del teclado de la consola para seleccionar el canal deseado.	Comprobar que se activa la radio VHF y se muestra en el display la información del nuevo canal y grupo seleccionado: VHF LINEA X TREN XX



		CANAL=X LIN=XX,XX,XX Radiotelefonía
2	Realizar una llamada en el nuevo canal seleccionado.	Comprobar que se activa el indicador "TX" de la consola, y que se escucha el audio en los terminales autorizados para el grupo de línea o depósito seleccionado.

Tabla 76: Caso de prueba CP\_RADIOTEL-04

Identificador	CP_RADIOTEL-05	
Descripción	Realizar llamada de radiotelefonía.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Modo radiotelefonía seleccionado.		
Paso	Procedimiento	Resultado esperado
1	Descolgar el microteléfono, pulsar PTT y hablar.	Comprobar que se activa el indicador “TX” de la consola, y que se escucha el audio en los terminales autorizados para el grupo de línea o depósito seleccionado si se está en TETRA, o en el canal correspondiente en VHF. Comprobar que no se escucha el audio en los terminales no autorizados.

Tabla 77: Caso de prueba CP\_RADIOTEL-05

Identificador	CP_RADIOTEL-06	
Descripción	Recibir llamada radiotelefonía.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Modo radiotelefonía seleccionado.		
Paso	Procedimiento	Resultado esperado
1	Realizar una llamada al grupo o canal seleccionado desde el Puesto de Mando (o desde el simulador del Puesto de Mando).	Comprobar que, si nuestro terminal está autorizado para el canal seleccionado en VHF, o en TETRA para el grupo de línea o depósito seleccionado: Se activa el indicador “RX” de la consola; se muestra el ID de la



		<p>llamada en la segunda línea del display; se recibe la llamada y se escucha el audio por el altavoz de la consola.</p> <p>Comprobar que no se recibe la llamada ni se escucha el audio en los terminales no autorizados.</p>
--	--	--

Tabla 78: Caso de prueba CP\_RADIOTEL-06

Identificador	CP_RADIOTEL-07	
Descripción	Manos libres en radiotelefonía.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Modo radiotelefonía seleccionado.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Manos Libres”.	Comprobar que en el display se muestra el indicador “(ML)” al lado del modo activo y que el led “Teléfono” parpadea.
2	Hablar directamente para realizar una llamada al grupo o canal seleccionado sin levantar el microteléfono.	Comprobar que se activa el indicador “TX” de la consola, y que se escucha el audio en los terminales autorizados para el canal seleccionado en VHF, o en TETRA para el grupo de línea o depósito seleccionado.  Comprobar que no se escucha el audio en los terminales no autorizados.

Tabla 79: Caso de prueba CP\_RADIOTEL-07

Identificador	CP_RADIOTEL-08	
Descripción	Modo interconexión usando TETRA.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio TETRA activa.		
Paso	Procedimiento	Resultado esperado
1	Con la radio TETRA activa y un grupo (línea o depósito) seleccionado, enviar un mensaje SDS de Intercomunicación desde el	Comprobar que se activa el indicador “INT” de la consola y todas las llamadas de grupo recibidas por terminales



	Puesto de Mando (o desde el simulador).	autorizados se escuchan en el recinto de pasajeros (o en el simulador de megafonía).
2	Pulsar la tecla “Radiotelefonía”.	Comprobar que se vuelve al modo al modo normal de funcionamiento.

Tabla 80: Caso de prueba CP\_RADIOTEL-08

Identificador	CP_RADIOTEL-09	
Descripción	Modo interconexión usando VHF.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio VHF activa.		
Paso	Procedimiento	Resultado esperado
1	Con la radio VHF activa y un canal seleccionado, pulsar la tecla “INT”.	Comprobar que: se activa el indicador “INT” de la consola; aparece en la cuarta línea del display “Int. LINEA XX TREN YYY”; todas las llamadas recibidas por el canal seleccionado se escuchan en el recinto de pasajeros (o en el simulador de megafonía).
2	Pulsar la tecla “Radiotelefonía”.	Comprobar que se vuelve al modo al modo “normal” de funcionamiento.

Tabla 81: Caso de prueba CP\_RADIOTEL-09

Identificador	CP_RADIOTEL-10	
Descripción	Modo local usando TETRA.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio TETRA activa. Modo radiotelefonía activado.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Radio Local” del teclado de la consola.	Comprobar que se activa el modo Radio Local y aparecen en el display el indicador “Radio Local”.
2	Con la radio TETRA activa y un grupo (depósito) seleccionado, descolgar el microteléfono, pulsar PTT y hablar.	Comprobar que se activa el indicador “TX” de la consola, y que se escucha el audio en los terminales TETRA (sin uso de la





		infraestructura) que tengan este modo activado. Comprobar que no se escucha el audio en los terminales no autorizados.
3	Cambiar el grupo a un grupo de línea y repetir los pasos anteriores.	Comprobar que no es posible activar el modo Radio Local.

Tabla 82: Caso de prueba CP\_RADIOTEL-10

Identificador	CP_RADIOTEL-11	
Descripción	Desactivación automática del modo radio local en TETRA.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio TETRA activa. Modo radiotelefonía activado.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Radio Local” del teclado de la consola.	Comprobar que se activa el modo Radio Local y aparecen en el display el indicador “Radio Local”.
2	Con la radio TETRA activa y un grupo de línea o depósito seleccionado, descolgar el microteléfono, pulsar PTT y hablar.	Comprobar que se activa el indicador “TX” de la consola, y que se escucha el audio en los terminales TETRA (sin uso de la infraestructura) que tengan este modo activado. Comprobar que no se escucha el audio en los terminales no autorizados.
3	Permanecer 30 segundos sin interaccionar con elementos de la consola.	Comprobar que el modo radio local se desactiva automáticamente al pasar los 30 segundos.

Tabla 83: Caso de prueba CP\_RADIOTEL-11

Identificador	CP_MEGAFO-01	
Descripción	Megafonía cabina.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Megafonía Cabina”.	Comprobar que se activa el indicador “MGF” en la consola y que en el display aparece



		"Megafonía cabina" la línea correspondiente.
2	Descolgar el teléfono, pulsar PTT y hablar.	Comprobar que se activa el indicador "TX" de la consola y que se escucha audio en la otra cabina (o en el simulador de megafonía).

Tabla 84: Caso de prueba CP\_MEGAFO-01

Identificador	CP_MEGAFO-02	
Descripción	Anulación del audio del panel frontal en megafonía cabina al descolgar el microteléfono.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio en modo megafonía cabina.		
Paso	Procedimiento	Resultado esperado
1	Realizar una transmisión de audio (simulando Puesto de Mando) a la consola en megafonía cabina.	Comprobar que existe audio en el altavoz de la consola.
2	Descolgar el microteléfono.	Comprobar que se activa el indicador “Teléfono” y que el audio deja de escucharse a través del altavoz y pasa a escucharse a través del microteléfono.

Tabla 85: Caso de prueba CP\_MEGAFO-02

Identificador	CP_MEGAFO-03	
Descripción	Megafonía público	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Megafonía público”.	Comprobar que se activa el indicador “MGF” en la consola y que en el display aparece “Megafonía público” en la línea correspondiente.
2	Descolgar el teléfono, pulsar PTT y hablar.	Comprobar que se activa el indicador “TX” de la consola y que se escucha audio en el sistema de megafonía del recinto de pasajeros (o en el simulador de megafonía).

Tabla 86: Caso de prueba CP\_MEGAFO-03



Identificador	CP_MEGAFO-04	
Descripción	Megafonía timbre.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida.		
Paso	Procedimiento	Resultado esperado
1	Activar un tirador de emergencia de la cabina de pasajeros (o utilizar el simulador de megafonía).	Comprobar que se activa el modo “Megafonía timbre” en la consola y que se activa el indicador “MGF” en la consola y que en el display aparece “Megafonía timbre” en la línea correspondiente.
2	Descolgar el teléfono, pulsar PTT y hablar para atender la llamada de megafonía timbre.	Comprobar que se activa el indicador “TX” de la consola y que se escucha audio en el sistema de megafonía del tirador de alarma (o en el simulador de megafonía).
3	Cancelar esta llamada, pulsar “Radiotelefonía” o conmutar a otro tipo de megafonía (megafonía cabina o megafonía público).	Comprobar que se conmuta al método seleccionado.
4	Retornar al modo “Megafonía timbre” pulsando la tecla “Megafonía timbre”.	Caso 1: si el tirador de emergencia sigue activado, comprobar que se vuelve a activar la comunicación inicial y es posible volver a realizar el Paso 2. Caso 2: si el tirador de alarma de emergencia se ha desactivado, comprobar que no se puede volver a activar la comunicación inicial.

Tabla 87: Caso de prueba CP\_MEGAFO-04

Identificador	CP_MESTADO-01		
Descripción	Envío de mensaje “Aviso a jefe de circulación”.		
Prerrequisitos			
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio activa TETRA.			
Paso	Procedimiento	Resultado esperado	
1	Pulsar la tecla “Aviso a jefe de circulación” (6) del teclado de la consola para enviar el mensaje al Puesto de Mando.	Comprobar que aparece la siguiente pantalla en el display: Lxx Nxx Ag xxxxx Grupo línea x	



		Enviar Av jefe circulación?
2	Pulsar la tecla Aceptar para enviar el mensaje.	Comprobar que aparece la siguiente pantalla en el display: Lxx Nxx Ag xxxxx Grupo línea x Radiotelefonía Enviando mensaje
3	Esperar a que termine de enviarse el mensaje.	Comprobar, que una vez enviado el mensaje, aparece durante cinco segundos la pantalla: Lx Nxx Ag xxxxx Grupo línea x Radiotelefonía Mensaje enviado Comprobar que el mensaje se recibe correctamente en el Puesto de Mando (o en el simulador).
4	Repetir el paso 1, y posteriormente pulsar "Borrar" para anular el envío del mensaje.	Comprobar que la consola vuelve al estado original y no se envía el mensaje.

Tabla 88: Caso de prueba CP\_MESTADO-01

Identificador	CP_MESTADO-02		
Descripción	Envío de mensaje “Rebase de punto de parada”.		
Prerrequisitos			
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio activa TETRA.			
Paso	Procedimiento	Resultado esperado	
1	Pulsar la tecla “Rebase punto de parada” (7) del teclado de la consola para enviar el mensaje al Puesto de Mando.	Comprobar que aparece la siguiente pantalla en el display: Lxx Nxx Ag xxxxx Grupo línea x Enviar Rebase parada?	
2	Pulsar la tecla Aceptar para enviar el mensaje.	Comprobar que aparece la siguiente pantalla en el display: Lxx Nxx Ag xxxxx Grupo línea x Radiotelefonía Enviando mensaje	
3	Esperar a que termine de enviarse el mensaje.	Comprobar, que una vez enviado el mensaje, aparece durante cinco segundos la pantalla:	



		<p>Lxx Nxx Ag xxxxx</p> <p>Grupo línea x</p> <p>Radiotelefonía</p> <p>Mensaje enviado</p> <p>Comprobar que el mensaje se recibe correctamente en el Puesto de Mando (o en el simulador).</p>
4	Repetir el paso 1, y posteriormente pulsar “Borrar” para anular el envío del mensaje.	Comprobar que la consola vuelve al estado original y no se envía el mensaje.

Tabla 89: Caso de prueba CP\_MESTADO-02

Identificador	CP_MESTADO-03		
Descripción	Envío de mensaje “Señal en rojo”.		
Prerrequisitos			
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio activa TETRA.			
Paso	Procedimiento	Resultado esperado	
1	Pulsar la tecla “Señal en rojo” (8) del teclado de la consola para enviar el mensaje al Puesto de Mando.	Comprobar que aparece la siguiente pantalla en el display: Lxx Nxx Ag xxxxx Grupo línea x Enviar Señal en rojo?	
2	Pulsar la tecla Aceptar para enviar el mensaje.	Comprobar que aparece la siguiente pantalla en el display: Lxx Nxx Ag xxxxx Grupo línea x Radiotelefonía Enviando mensaje	
3	Esperar a que termine de enviarse el mensaje.	Comprobar, que una vez enviado el mensaje, aparece durante cinco segundos la pantalla: Lxx Nxx Ag xxxxx Grupo línea x Radiotelefonía Mensaje enviado Comprobar que el mensaje se recibe correctamente en el Puesto de Mando (o en el simulador).	
4	Repetir el paso 1, y posteriormente pulsar “Borrar” para anular el envío del mensaje.	Comprobar que la consola vuelve al estado original y no se envía el mensaje.	

Tabla 90: Caso de prueba CP\_MESTADO-03



Identificador	CP_MESTADO-04		
Descripción	Envío de mensaje “Quedan viajeros en andén”.		
Prerrequisitos			
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio activa TETRA.			
Paso	Procedimiento	Resultado esperado	
1	Pulsar la tecla “Quedan viajeros en andén” (9) del teclado de la consola para enviar el mensaje al Puesto de Mando.	Comprobar que aparece la siguiente pantalla en el display: Lxx Nxx Ag xxxxx Grupo línea x Enviar Viajeros en andén?	
2	Pulsar la tecla Aceptar para enviar el mensaje.	Comprobar que aparece la siguiente pantalla en el display: Lxx Nxx Ag xxxxx Grupo línea x Radiotelefonía Enviando mensaje	
3	Esperar a que termine de enviarse el mensaje.	Comprobar, que una vez enviado el mensaje, aparece durante cinco segundos la pantalla: Lxx Nxx Ag xxxxx Grupo línea x Radiotelefonía Mensaje enviado Comprobar que el mensaje se recibe correctamente en el Puesto de Mando (o en el simulador).	
4	Repetir el paso 1, y posteriormente pulsar “Borrar” para anular el envío del mensaje.	Comprobar que la consola vuelve al estado original y no se envía el mensaje.	

Tabla 91: Caso de prueba CP\_MESTADO-04

Identificador	CP_MESTADO-05
Descripción	Envío de mensaje “Abandono cabina”.
Prerrequisitos	
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio activa TETRA.	



Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Abandono cabina” del teclado de la consola para enviar el mensaje al Puesto de Mando.	Comprobar que aparece la siguiente pantalla en el display: Lxx Nxx Ag xxxxx Grupo línea x Enviar Abandono cabina?
2	Pulsar la tecla Aceptar para enviar el mensaje.	Comprobar que aparece la siguiente pantalla en el display: Lxx Nxx Ag xxxxx Grupo línea x Radiotelefonía Enviando mensaje
3	Esperar a que termine de enviarse el mensaje.	Comprobar, que una vez enviado el mensaje, aparece durante cinco segundos la pantalla: Lxx Nxx Ag xxxxx Grupo línea x Radiotelefonía Mensaje enviado Comprobar que el mensaje se recibe correctamente en el Puesto de Mando (o en el simulador).
4	Repetir el paso 1, y posteriormente pulsar “Borrar” para anular el envío del mensaje.	Comprobar que la consola vuelve al estado original y no se envía el mensaje.

Tabla 92: Caso de prueba CP\_MESTADO-05

Identificador	CP_MESTADO-06		
Descripción	Envío de mensaje “Puerta abierta”.		
Prerrequisitos			
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio activa TETRA.			
Paso	Procedimiento	Resultado esperado	
1	Pulsar la tecla “Puerta abierta” del teclado de la consola para enviar el mensaje al Puesto de Mando.	Comprobar que aparece la siguiente pantalla en el display: Lxx Nxx Ag xxxxx Grupo línea x Enviar Puerta abierta?	
2	Pulsar la tecla Aceptar para enviar el mensaje.	Comprobar que aparece la siguiente pantalla en el display:  Lxx Nxx Ag xxxxx Grupo línea x	



		Radiotelefonía Enviando mensaje
3	Esperar a que termine de enviarse el mensaje.	Comprobar, que una vez enviado el mensaje, aparece durante cinco segundos la pantalla: Lxx Nxx Ag xxxxx Grupo línea x Radiotelefonía Mensaje enviado Comprobar que el mensaje se recibe correctamente en el Puesto de Mando (o en el simulador).
4	Repetir el paso 1, y posteriormente pulsar “Borrar” para anular el envío del mensaje.	Comprobar que la consola vuelve al estado original y no se envía el mensaje.

Tabla 93: Caso de prueba CP\_MESTADO-06

Identificador	CP_MESTADO-07	
Descripción	Envío de mensajes desde VHF.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Radio activa VHF.		
Paso	Procedimiento	Resultado esperado
1	Pulsar alguna de las teclas de envío de mensajes del teclado de la consola para enviar el mensaje al Puesto de Mando.	Comprobar que las teclas no realizan función ninguna y no se envía el mensaje.

Tabla 94: Caso de prueba CP\_MESTADO-07

Identificador	CP_MENUCON-01		
Descripción	Menú “Datos de circulación”.		
Prerrequisitos			
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Consola en pantalla principal.			
Paso	Procedimiento		Resultado esperado
1	Pulsar la tecla “Menú” para		Comprobar el funcionamiento





	<p>acceder al menú de la consola. Se accede a los submenús con la tecla “Aceptar” y se retrocede al menú anterior con “Borrar”. Usar las teclas “Subir” y “Bajar” para mover el cursor.</p> <p>Seleccionar la opción “1. Datos de circulación”.</p>	<p>correcto del menú “Datos de circulación” accediendo al mismo y modificando valores:</p> <p>&gt;1. Datos de circulación</p> <ol style="list-style-type: none"> <li>1. Numero conductor?</li> <li>2. Numero de tren?</li> <li>3. Numero de línea?</li> </ol> <p>Comprobar que se envían correctamente los datos de circulación al sistema.</p>
--	---	---

Tabla 95: Caso de prueba CP\_MENUCON-01

Identificador	CP_MENUCON-02	
Descripción	Menú “Establecer radio”.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Consola en pantalla principal.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Menú” para acceder al menú de la consola. Se accede a los submenús con la tecla “Aceptar” y se retrocede al menú anterior con “Borrar”. Usar las teclas “Subir” y “Bajar” para mover el cursor. Seleccionar la opción “2. Establecer radio”.	Comprobar el funcionamiento correcto del menú “Establecer radio” accediendo al mismo y seleccionando las opciones: 2. Establecer radio 1. Radio VHF 2. Radio TETRA

Tabla 96: Caso de prueba CP\_MENUCON-02

Identificador	CP_MENUCON-03	
Descripción	Menú “Buzón TETRA”.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Consola en pantalla principal. Radio TETRA activa.		
Paso	Procedimiento	Resultado esperado
1	Pulsar la tecla “Menú” para acceder al menú de la consola. Se accede a los submenús con la tecla “Aceptar” y se retrocede al menú anterior con “Borrar”. Usar las	Comprobar el funcionamiento correcto del menú “Buzón TETRA” accediendo al mismo y modificando valores: 3. Buzón TETRA



	teclas “Subir” y “Bajar” para mover el cursor. Seleccionar la opción “3. Buzón TETRA”	1. Leer Mensajes 2. Borrar Mensajes
2	Acceder al submenú “Borrar mensajes”.	Comprobar que se almacena un máximo de cinco mensajes en memoria y se pueden borrar.
3	Acceder al submenú “Leer mensajes”.	Comprobar que los mensajes tienen un tamaño máximo de 140 caracteres.
4	Establecer radio activa VHF.	Comprobar que no se puede acceder al menú “Buzón TETRA”.

Tabla 97: Caso de prueba CP\_MENUCON-03

Identificador	CP_MENUCON-04	
Descripción	Menú “Grupos fonía”.	
Prerrequisitos		
Todos los elementos conectados correctamente. Situación de rack en servicio. Situación de cabina activa o consola encendida. Consola en pantalla principal.		
Paso	Procedimiento	Resultado esperado
1	Pulsar las teclas “Canal +” o “Canal -” para acceder al submenú de grupos de fonía. Se accede a los submenús con la tecla “Aceptar” y se retrocede al menú anterior con “Borrar”. Usar las teclas “Subir” y “Bajar” para mover el cursor.	Comprobar el funcionamiento correcto del submenú “Grupos de fonía” accediendo al mismo y modificando valores:  1. Grupo línea 2. Grupo deposito
2	Acceder al submenú de grupos de fonía y no realizar ninguna acción	Comprobar que al pasar 1 minuto la consola vuelve a mostrar la pantalla principal

Tabla 98: Caso de prueba CP\_MENUCON-04

## 8. Presupuesto

En esta sección se va a realizar un análisis del presupuesto para este proyecto. Se evaluará de manera teórica, a través de la una herramienta que evalúa los costes según el modelo de COCOMO y luego se compararán con los costes reales que generó este desarrollo.

### 8.1. Planificación y costes

Para este proyecto se realizó una planificación inicial que cumpliera con los plazos especificados por el cliente, y que incluyera todas las fases del desarrollo y pruebas del sistema.

El cronograma con la planificación original es el siguiente.

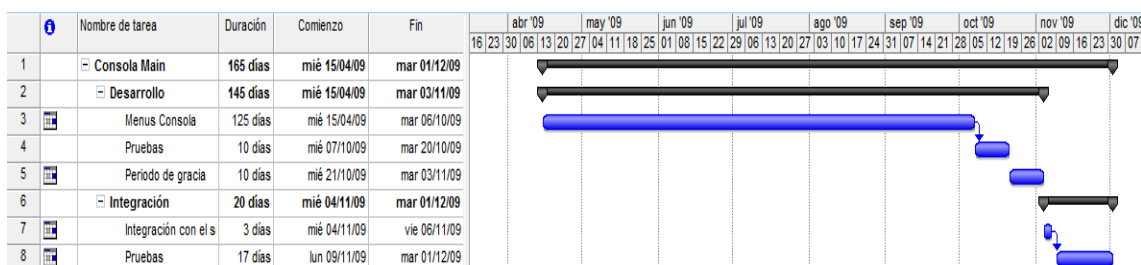


Ilustración 12: Planificación inicial

Por contra, la aplicación se desarrollo realmente según vemos en el siguiente cronograma.

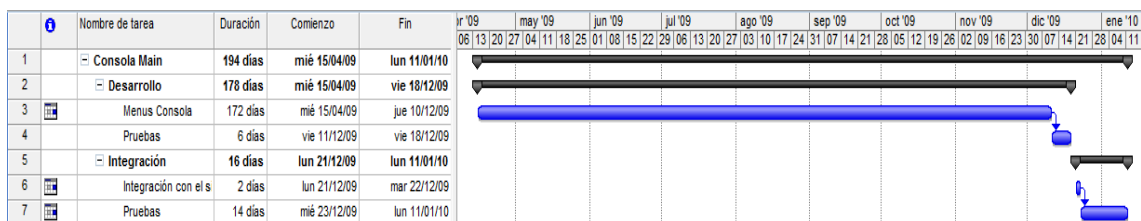


Ilustración 13: Planificación real

Como podemos observar, dados estos dos gráficos, el desarrollo de la aplicación no cumplió con las expectativas previstas. La primera fase consistente en el desarrollo del sistema vemos que se prolongó más de lo previsto inicialmente. Esto principalmente fue debido a los cambios producidos en los requisitos de la aplicación.

Por otro lado, vemos como la fase de integración y pruebas pudo reducirse respecto a la planificación inicial, puesto que el desarrollo de la aplicación estaba ya bastante bien acabado. Como se comentó en los puntos anteriores, hubo diversos fallos que surgieron durante esta fase, y éstos no estaban contemplados en la planificación inicial. No obstante, se contrarrestaron con el desarrollo de la aplicación que necesito menos retoques de los esperados en un principio.

Todas estas variaciones, como es lógico, repercutirán finalmente en los costes finales del producto.

Para establecer los costes asociados al proyecto, únicamente podré especificar el coste en horas por persona, debido a criterios de confidencialidad con la empresa. Además de esto, y por el mismo motivo, tampoco se podrán especificar los costes asociados a los componentes hardware, simuladores y demás sistemas que se necesitan para el desarrollo de este proyecto.

Por lo tanto, teniendo en cuenta lo especificado en el cronograma asociado al tiempo para el desarrollo de la aplicación, inicialmente, se estimó lo siguiente:

Concepto	Días
145 días de desarrollo.	145 días
20 días de integración y pruebas.	20 días
Total	165 días (165 días x 8 horas / día = 1320 horas)

Tabla 99: Horas previstas asociadas al proyecto

Sin embargo, finalmente el resultado final de la planificación de este desarrollo fue el siguiente:

Concepto	Días
145 días de desarrollo.	178 días
20 días de integración y pruebas.	16 días
Total	194 días (194 días x 8 horas / día = 1552 horas)

Tabla 100: Horas reales asociadas al proyecto

Por lo tanto, se puede concluir que el coste total del proyecto asociado a los gastos de personal asciende a **1552 horas por hombre**.

## 8.2. Análisis según COCOMO

En esta sección se va a realizar un análisis según la metodología COCOMO (Constructive Cost Model) [8].

Para este modelo matemático necesitamos introducir varias variables para obtener los valores predictivos. El número de estas variables dependerá del nivel de detalle y aproximación que se pretende tener del sistema.



Para este producto, el nivel que se estudiará es el intermedio, para el cual se establecen las variables que veremos a continuación.

Primeramente se establecen los factores de escala, siendo los asociados al proyecto los siguientes.

Factor	Valor	Razón
Precedentes	Bajo (4,96).	No hay mucha experiencia en proyectos similares.
Flexibilidad del desarrollo	Bajo (4,05).	Baja flexibilidad, existen diversas restricciones.
Arquitectura	Bajo (5,65).	No existe mucho conocimiento de la arquitectura empleada.
Cohesión del equipo	Alto (2,19).	Alta integración del equipo de trabajo.
Madurez del proceso	Alto (3,12).	Se establece como alto según CMMI.

Tabla 101: Factores de escala del proyecto

Por otro lado tenemos los atributos del proyecto, los cuales se dividen en distintos entornos. Estos entornos se especifican a continuación junto con sus atributos correspondientes:

- **Atributos de software**

Los atributos del hardware son los siguientes:

- RELY: garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto.
- DATA: tamaño de la base de datos en relación con el tamaño del programa.
- CPLX: representa la complejidad del producto.
- RUSE: Reusabilidad del código.
- DOCU: documentación del proyecto.

Los valores establecidos para este proyecto son los especificados en esta tabla.

Atributo	Valor	Razón
RELY	Alto (1,10).	Es necesaria una alta fiabilidad del sistema.
DATA	Bajo (0,90).	La base de datos es muy pequeña.



CPLX	Normal (1,00).	La complejidad de la aplicación es normal.
RUSE	Baja (0,95).	No habrá mucha reusabilidad del código.
DOCU	Baja (0,91).	Poca documentación de la aplicación.

Tabla 102: Atributos de software

- **Atributos de hardware**

Los atributos del hardware son los siguientes:

- TIME: limitaciones en el porcentaje del uso de la CPU.
- STOR: limitaciones en el porcentaje del uso de la memoria.
- PVOL: volatilidad del proyecto.

Los valores establecidos para este proyecto son los especificados en esta tabla.

Atributo	Valor	Razón
TIME	Muy alto (1,11).	Las restricciones en el tiempo son muy importantes.
STOR	Muy alto (1,05).	La memoria existente es muy limitada.
PVOL	Normal (1,00).	La volatilidad es normal.

Tabla 103: Atributos de hardware

- **Atributos de personal**

Los atributos del personal son los siguientes:

- ACAP: calificación de los analistas.
- AEXP: experiencia del personal en aplicaciones similares.
- PCAP: calificación de los programadores.
- PEXP: experiencia del personal en la plataforma.
- LEXP: experiencia en el lenguaje de programación a usar.
- PCON Continuidad del a gente involucrada en el proyecto.

Los valores establecidos para este proyecto son los especificados en esta tabla.

Atributo	Valor	Razón
ACAP	Alto (0,85).	Cualificación de los analistas está por encima del 75%.
AEXP	Normal (1,00).	La experiencia es normal.
PCAP	Alto (0,88).	Cualificación de los programadores está por



		encima del 75%.
PEXP	Bajo (0,88).	La experiencia en la plataforma es baja.
LEXP	Muy bajo (1,20).	La experiencia con el lenguaje de programación es casi nula.
PCON	Alto (0'90).	Alta continuidad del personal del proyecto.

Tabla 104: Atributos de personal

- **Atributos de proyecto**

Los atributos del proyecto son los siguientes:

- SITE: Entorno que comparte el equipo de desarrollo
- TOOL: uso de herramientas de desarrollo de software.
- SCED: limitaciones en el cumplimiento de la planificación.

Los valores establecidos para este proyecto son los especificados en esta tabla.

Atributo	Valor	Razón
SITE	Muy alto (0,78).	El equipo de trabajo comparte la misma área y poseen gran ancho de banda.
TOOL	Normal (1,00).	Se usan diversas aplicaciones CASE y de desarrollo para la implementación del producto.
SCED	Alto (1,00).	Hay altas limitaciones de tiempo para el desarrollo.

Tabla 105: Atributos de proyecto

### 8.3. Análisis comparativo de costes

Debido a problemas de confidencialidad con los costes asociados al proyecto como se explico anteriormente, ya que se trata de un proyecto real, nos centraremos en el análisis de los costes asociados al personal, especificándolo en unidades de horas por hombre.

Como se puede observar en la planificación, inicialmente la previsión fue de 1320 horas por hombre, sin embargo, finalmente, debido a los imprevistos y cambios en las especificaciones del producto, el total al que ascendió el coste horas por hombre fue de 1552 horas aproximadamente.



Por lo tanto, finalmente, el coste total es casi un 20% mayor del estimado en un inicio, en concreto un 17,6% más.

No obstante, esto es en lo referente a los costes asociados a los empleados, sin embargo, el coste asociado a la consola, incluyendo los componentes hardware, se incremento en total en casi un 22%.

Es por lo tanto muy importante realizar una planificación y unos presupuestos que tengan en cuenta multitud de factores e inconvenientes que puedan surgir, puesto que finalmente es muy probable que se produzcan imprevistos asociados al proyecto, o lo que es más difícil de prever, los ajenos al mismo.



## 9. Conclusiones

Este trabajo ha sido mi primer gran proyecto real tras la universidad.

Ha sido una experiencia realmente enriquecedora. Me ha servido para aprender multitud de cosas nuevas y a poner en práctica otras muchas que he ido viendo y aprendiendo a lo largo de los años que he estado estudiando esta carrera.

Me ha servido para darme cuenta de lo mucho que me gusta lo que hago, lo interesante que es producir algo nuevo, diferente y enfrentarme a multitud de retos con la presión de ser la vida real, puesto que aquí ya no me jugaba un aprobado, si no mucho más, y no sólo yo, si no todo el equipo del que formaba parte.

También me ha sorprendido lo importante que son ciertas cosas, que a priori durante la carrera no le daba la importancia que se merecían, y que ahora agradezco haber aprendido, o aunque solo sea, tener ciertas nociones de alguna determinada materia.

No obstante, han sido unos meses complicados, pues ya nada más empezar tuve que enfrentarme a multitud de dificultades. Inicialmente todo era nuevo para mí, el equipo de trabajo, las tecnologías a usar o la manera de trabajar de la gente de mi entorno. Tuve que adaptarme rápidamente, y empezar a aprender cosas tan básicas para el desarrollo de la aplicación como podía ser el lenguaje con el que se iba a implementar.

Durante todo este periodo he podido realizar muchas tareas dentro de este proyecto. Todas estas tareas abarcan una amplia parte de los conocimientos que he ido adquiriendo a lo largo de los años de carrera. Las más importantes a destacar son las siguientes:

- Programación en Python. He tenido que programar con un lenguaje completamente nuevo. Gracias a la amplia experiencia obtenida en la universidad con Java, el paradigma de la orientación a objetos lo tenía bien asimilado y pude rápidamente adaptarme al nuevo lenguaje.
- Manipulación de bases de datos. Esta parte del proyecto venía fundamentalmente en la fase de pruebas, puesto que muchas pruebas se realizaban manipulando la base de datos del sistema.
- Uso de herramientas CASE. Para este proyecto se han usado diversas herramientas CASE. Con estas herramientas se generó documentación útil para el desarrollo de la aplicación.
- Ingeniería del software. Este ha sido un gran proyecto, en el que he podido comprender y asimilar el ciclo de vida del software, y lo importante que es la documentación y análisis del producto que se va a desarrollar para realizarlo convenientemente.

Y por último, para concluir, podemos afirmar que se han cumplido todos los objetivos de este proyecto.

- Desarrollo del sistema: El sistema se desarrollo correctamente cumpliendo con los requisitos especificados por el cliente, incluyendo además mejoras que incrementaron el valor del producto.
- Implantación del sistema: El sistema se pudo instalar en los trenes de una manera satisfactoria, integrándose perfectamente con los sistemas existentes en los trenes, y ofreciendo la funcionalidad deseada
- Alta calidad del producto: Se ha creado un producto de excelente calidad, cumpliendo con las expectativas y los niveles de fiabilidad esperados, los cuales eran de vital importancia para el cliente.



Ilustración 14: Tren de la serie 2000



## 10. Bibliografía

- [1] Metro de Madrid S. A.: <http://www.metromadrid.es/es/index.html>  
Visitado el 8 de junio de 2010.
- [2] Amper S. A.: <http://www.amper.es/index.cfm?lang=sp>  
Visitado el 8 de junio de 2010.
- [3] Tetra: <http://www.tetra-association.com/>  
Visitado el 8 de junio de 2010.
- [4] CAF, trenes serie 2000:  
<http://www.caf.es/caste/productos/proyecto.php?cod=1&id=196&sec=desc>  
Visitado el 8 de junio de 2010.
- [5] Python: <http://www.python.org/>  
Visitado el 8 de junio de 2010.
- [6] GLib: <http://library.gnome.org/devel/glib/>  
Visitado el 8 de junio de 2010.
- [7] D-Bus: <http://www.freedesktop.org/wiki/Software/dbus>  
Visitado el 8 de junio de 2010.
- [8] COCOMO: [http://sunset.usc.edu/csse/research/COCOMOII/cocomo\\_main.html](http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html)  
Visitado el 8 de junio de 2010.

## Glosario

- **Consola:** Terminal con el que interactuará el maquinista. Consta de un microteléfono, un sistema de manos libres, un display gráfico y un teclado.
- **Rack:** Equipo físico en el que van instalados los sistemas de comunicaciones (radios VHF y TETRA) así como el equipo en el que se ejecutará toda la lógica de las diversas aplicaciones necesarias para el correcto funcionamiento del sistema.
- **RackDaemon:** Demonio que se ejecuta en el rack de comunicaciones con el que la Consola interactúa a través de un bus de datos. Ofrece funcionalidad al programa de la consola, así como le informa de eventos generados por el sistema.
- **VHF (*Very High Frequency*):** Es la banda del espectro electromagnético que ocupa el rango de frecuencias de 30 MHz a 300 MHz. Es la frecuencia usada en la radiotelefonía, siendo ésta una tecnología que posibilita la transmisión de señales mediante la modulación de ondas electromagnéticas.
- **TETRA (*Terrestrial Trunked Radio*):** Es un estándar definido por el Instituto Europeo de Estándares de Telecomunicación. Este estándar define un sistema móvil digital de radio y nace por decisión de la Unión Europea con el objeto de unificar diversas alternativas de interfaces de radio digitales para la comunicación entre los profesionales de distintos sectores.
- **PTT (*Push To Talk*):** Es un método para hablar en líneas semi-duplex de comunicación, apretando un botón para transmitir y liberándolo para recibir. Este tipo de comunicación permite llamadas de tipo uno-a-uno o bien uno-a-varios (llamadas de grupos).
- **Inverter:** Señal de puesta en marcha de los trenes.
- **COCOMO:** El Modelo Constructivo de Costes (o COCOMO, por su acrónimo del inglés CONstructive COst MOdel) es un modelo matemático de base empírica utilizado para estimación de costes de software.